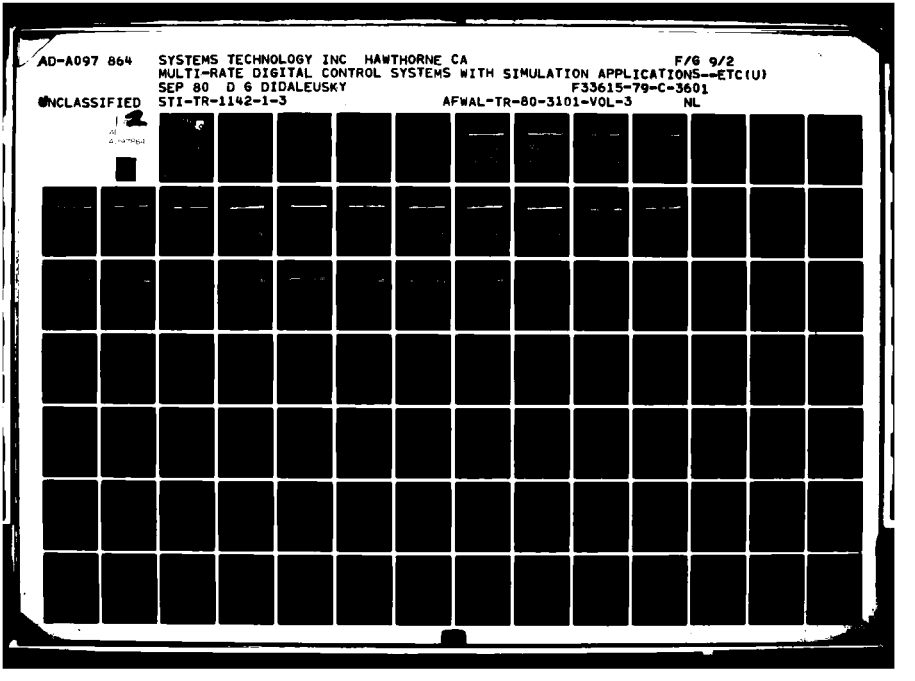
AD-A097 864 SYSTEMS TECHNOLOGY INC HAWTHORNE CA F/G 9/2
MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS--ETC(U)
SEP 80 D G DIDALEUSKY F33615-79-C-3601
UNCLASSIFIED STI-TR-1142-1-3 AFWAL-TR-80-3101-VOL-3 NL



AFWAL-TR-80-3101
Volume III

LEVEL III

6

AD A097864

MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS

Volume III: Source Listings

DENNIS G. J. DIDALEUSKY
FLIGHT DYNAMICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433

SEPTEMBER 1980

TECHNICAL REPORT AFWAL-TR-80-3101 VOL III
Final Report — January 1979-May 1980

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

DENNIS G. J. DIDALEUSKY, Captain, USAF
Project Engineer
Control Dynamics Branch
Flight Control Division

R. O. ANDERSON, Chief
Control Dynamics Branch
Flight Control Division

FOR THE COMMANDER

ROBERT C. ETTINGER, Colonel, USAF
Chief, Flight Control Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIGC, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/27 March 1981 — 510

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
AFWAL-TR-80-3101 - Vol III	AD-A097864	

4. TITLE (and Subtitle)
MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS. Volume III: Source Listings.

5. TYPE OF REPORT & PERIOD COVERED
Final Report.
29 Jan 79 — 29 May 80

6. PERFORMING ORG. REPORT NUMBER
STI-TR-1142-1-3

7. AUTHOR(s)
Dennis J. Didaleusky

8. CONTRACT OR GRANT NUMBER(s)
F33615-79-C-3601

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Systems Technology, Inc.
13766 South Hawthorne Boulevard
Hawthorne, California 90250

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Program Element 61102F,
Project 2304, Mathematics,
Task N3, Math. of Flt. Control

11. CONTROLLING OFFICE NAME AND ADDRESS
Flight Dynamics Laboratory
Air Force Wright Aeronautical Laboratories
Wright-Patterson AFB, Ohio 45433

12. REPORT DATE
September 1980

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution limited to U.S. Government Agencies only; Test and Evaluation; September 1980. Other requests for this document must be referred to AFWAL/FICC, Wright-Patterson AFB, OH 45433.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Control Systems	Frequency Response	Computational Delays	Servo Analysis
Sampled Data	Switch Decomposition	Tustin Transform	Closed-Loop Analysis
z-Transforms	Closed-Loop Systems	Simulation Error Analysis	
w-Domain	Linear Systems		
Multi-Rate Sampling	Root Locus		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The report is organized in three volumes. Volume I contains the theoretical developments as well as illustrative examples and case studies. Volume II describes two algorithms useful in the analysis of multi-rate systems, the DISCRET and tXCONV computer programs. Volume III contains the listings of all programs and subroutines which comprise DISCRET and TXCONV.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

340425

FOREWORD

The research described in this report was performed by Systems Technology, Inc., Hawthorne, California, under Air Force Contract F33615-79-C-3601. The Task Number N3, Mathematics of Flight Control, was under Project Number 2304, Mathematics. This work was directed by the Control Dynamics Branch, Flight Control Division, Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Paterson Air Force Base, Ohio. The work was administered by Captain Dennis G. J. Didaleusky.

Richard F. Whitbeck was the Systems Technology, Inc., Project Engineer under the direction of Duane McRuer.

The authors wish to express their appreciation to the Systems Technology publication staff for their efforts in preparing this three-volume report.

The authors also wish to express their thanks to Ms. Susan Riedel at Systems Technology, Inc., and to Captain Stanley Larimer and Dr. Robert Schwanz at the Flight Dynamics Laboratory for their appreciable efforts in reviewing the technical report.

This report is organized in three volumes. Volume I contains the theoretical developments as well as illustrative examples and case studies. Volume II describes two algorithms useful in the analysis of multi-rate systems, the DISCRET and TXCONV computer programs. Volume III contains the FORTRAN listings for these computer programs.

This report covers work performed from January 1979 through May 1980. The report was submitted by the authors in August 1980.

TABLE OF CONTENTS

RE: AFWAL-TR-80-3101, Volume III, Distribution Statement-
Approved for Public Release per Ms. Martha Kline, AFFDL/STINFO

```
C   ********************************************************************** 000001
C   ********************************************************************** 000002
C   ********************************************************************** 000003
C   ********************  PROGRAM DISCRET  ******************************* 000004
C   ********************************************************************** 000005
C   ********************************************************************** 000006
C   ********************************************************************** 000007
C                                                                          000008
C                                                                          000009
C----------------------------------------------------------------------    000010
      OVERLAY(DISCRET,0,0)                                                 000011
      PROGRAM DISCRET(INPUT,OUTPUT,TAPE5=INPUT,TAPE7=OUTPUT)               000012
      COMMON/TOTL12/CLNPOLY(51),CLDPOLY(51),CL2ERO(50,2),                  000013
     +CLPOLE(50,2),NCLZ,NCLP,CLK,CLNK,CLDK                                 000014
      CALL OVERLAY(7HDISCRET,22,0)                                         000015
      END                                                                  000016
C----------------------------------------------------------------------    000017
C                                                                          000018
C                                                                          000019
C                                                                          000020
C                                                                          000021
C   ********************************************************************** 000022
C   *                                                                    * 000023
C   *          *****  DESCRIPTION OF DISCRET PROGRAM  *****              * 000024
C   *                                                                    * 000025
C   *  THE DISCRET PROGRAM CONVERTS A GENERAL ANALOG OR CONTINUOUS       * 000026
C   *  TRANSFER FUNCTION IN THE S-PLANE , G(S), INTO A VALID             * 000027
C   *  DISCRETE DOMAIN - THE Z, W, OR W' PLANE.  DISCRETE CAN            * 000028
C   *  CACULATE THE STANDARD, DELAYED, OR ADVANCED DISCRETE              * 000029
C   *  TRANSFORM.  DATA HOLDS INCLUDING THE ZERO ORDER, FIRST            * 000030
C   *  ORDER, SECOND ORDER, AND SLEWER CAN BE INSERTED INTO              * 000031
C   *  THE TRANSFORMATION.                                               * 000032
C   *                                                                    * 000033
C   *  THE PROGRAM  TAKES THE PARTIAL FRACTION EXPANSION OF              * 000034
C   *  THE G(S) TRANSFER FUNCTION, INCLUDING THOSE ZEROS AND             * 000035
C   *  POLES INTRODUCED BY THE PARTICULAR DATA HOLD SELECTED             * 000036
C   *  (E.G., THE ZERO ORDER HOLD INTRODUCES A POLE AT S=0 AND           * 000037
C   *  THE SLEWER TWO POLES AT S=0).  THE W PLANE TRANSFORM OF           * 000038
C   *  THE INDIVIDUAL PARTIAL FRACTION EXPANSION TERMS ARE               * 000039
C   *  OBTAINED AND RECOMBINED TO GIVE THE NUMERATOR AND                 * 000040
C   *  DENOMINATOR POLYNOMIALS IN THE DISCRETE W DOMAIN.  THE            * 000041
C   *  ROOTS ARE  COMPUTED AND PRINTED ALONG WITH THE                    * 000042
C   *  NUMERATOR AND DENOMINATOR POLYNOMIALS.  DEPENDING ON THE          * 000043
C   *  OPTION SELECTED,  THE CORRESPONDING Z AND W' TRANSFORMS           * 000044
C   *  CAN BE CALCULATED USING THE BILINEAR TRANSFORMATION               * 000045
C   *                                                                    * 000046
C   *                      W = 2-1/Z+1                                   * 000047
C   *                                                                    * 000048
C   *  AND THE REALTIONSHIP BETWEEN THE W AND W' PLANES                  * 000049
C   *  EXPRESSED AS                                                      * 000050
C   *                                                                    * 000051
C   *       W = (T/2)*W' ,      T= SAMPLING TIME (SEC)                   * 000052
C   *                                                                    * 000053
C   *  THE POSITIVE TIME ADVANCE AND NEGATIVE TIME DELAY                 * 000054
C   *  ASSOCIATED WITH THE ADVANCED AND DELAYED DISCRETE                 * 000055
```

```
C     *  TRANSFORMS IS INSERTED INTO THE TRANSFORMATION                    *  000056
C     *  WHEN SPECIFIED BY THE INPUT DATA.                                 *  000057
C     *                                                                    *  000058
C     *  THE INPUT DATA FOR THE PROGRAM IS DIVIDED INTO THREE MAIN         *  000059
C     *  BLOCKS OF DATA.  THE FIRST BLOCK CONTAINS THE BASIC               *  000060
C     *  PARAMETERS THAT DEFINE THE S PLANE TRANSFER FUNCTION G(S)         *  000061
C     *  TO BE TRANSFORMED.  THIS DATA IS PLACED ON THE FIRST DATA         *  000062
C     *  CARD IN A FREE-FORMAT.  THE ALPHANUMERIC CODE FOR THE DATA        *  000063
C     *  HOLD AND TYPE OF DISCRETE TRANSFORM ARE INSERTED ON CARDS         *  000064
C     *  TWO AND THREE IN (A3) AND (A2) FORMAT RESPECTIVELY.               *  000065
C     *  THE FINAL BLOCK OF DATA IS AGAIN FREE-FORMAT.  THIS               *  000066
C     *  DATA STARTS ON DATA CARD FOUR AND CONSISTS OF THE ZEROS           *  000067
C     *  AND POLES OF THE S-PLANE SYSTEM G(S).                             *  000068
C     *                                                                    *  000069
C     *  THE OUTPUT FROM DISCRET IS BASICALLY DEVIDED INTO TWO MAIN        *  000070
C     *  SECTIONS.  THE FIRST SECTION DEALS WITH THE INPUT PARAMETERS      *  000071
C     *  FOR THE S-PLANE CONTINUOUS SYSTEM G(S) AND THOSE PARAMETERS       *  000072
C     *  ASSOCIATED WITH THE DISCRETE TRANSFORMATION.  THE SECOND          *  000073
C     *  SECTION CONTAINS THE RESULTS OF THE TRANSFORMATION PROCESS,       *  000074
C     *  THE DISCRETE TRANSFER FUNCTION G(Z), G(W), OR G(W').              *  000075
C     *                                                                    *  000076
C     *  DISCRET FIRST PRINTS THE PARAMETERS AND OPTIONS SELECTED FOR      *  000077
C     *  THE DISCRETE TRANSFORMATION (I.E., SAMPLING TIME, DATA HOLD       *  000078
C     *  SELECTED, TYPE OF DISCRETE TRANSFORM, AND THE TIME INCREMENT      *  000079
C     *  USED).  THIS IS FOLLOWED WITH A LIST OF THE S-PLANE ZEROS         *  000080
C     *  AND POLES OF G(S) INCLUDING THOSE INTRODUCED BY THE               *  000081
C     *  PARTICULAR DATA HOLD SELECTED.  THE PARTIAL FRACTION              *  000082
C     *  EXPANSION COEFFICIENTS FOR G(S) ARE PRINTED NEXT.  THE            *  000083
C     *  DISCRETE NUMERATOR AND DENOMINATOR POLYNOMIALS AND THEIR          *  000084
C     *  ASSOCIATED ZEROS AND POLES ARE THEN GIVEN FOR G(Z), G(W),         *  000085
C     *  OR G(W').                                                         *  000086
C     *                                                                    *  000087
C     ************************************************************************  000088
C                                                                             000089
C                                                                             000090
C                                                                             000091
C-------------------------------------------------------------------------    000092
CDECK ADUANZ                                                                  000093
*     OVERLAY(26,0)                                                           000094
*     PROGRAM ADUANZ                                                          000095
C-------------------------------------------------------------------------    000096
C                                                                             000097
C                                                                             000098
C                                                                             000099
C-------------------------------------------------------------------------    000100
      PROGRAM ADUANZ(INPUT,OUTPUT,TAPE5=INPUT,TAPE7=OUTPUT)                   000101
C-------------------------------------------------------------------------    000102
C                                                                             000103
C                                                                             000104
C                                                                             000105
C                                                                             000106
      DOUBLE RN(102),RD(102)                                                  000107
      DOUBLE BR(51),BI(51),UR(51),UI(51)                                      000108
      DOUBLE AA,A0,T                                                          000109
                                                                              000110
```

```
      DIMENSION MM(51)                                                  000111
      COMMON/ADUCZ/RM,RD,NRTM,NRTD,MM,UR,UI,BR,BI,T,NH,AM,A0,           000112
     +TXFORM,CPLR                                                       000113
      COMMON/TOTL12/CLMPOLY(51),CLDPOLY(51),CLZERO(50,2),               000114
     +CLPOLE(50,2),NCLZ,NCLP,CLK,CLNK,CLDK                              000115
C                                                                       000116
C                                                                       000117
C                                                                       000118
CINT  ********************************************************        000119
CINT  ******  READ SECTION FOR INTERACTIVE OPERATION  ******            000120
CINT  ********************************************************        000121
CINT                                                                    000122
CINT USER INPUT SECTION FOR SAMPLING TIME, D/A COUPLER OPTION,          000123
CINT DISCRETE TRANSFORM OPTION, AND TIME DELAY/ADVANCED OPTION.         000124
CINT                                                                    000125
 2060 CONTINUE                                                          000126
CINT  PRINT%,'ENTER SAMPLING TIME IN SECONDS'                           000127
CINT  PRINT%,'TSAMP (SEC)='                                             000128
CINT  READ(5,%)T                                                        000129
CINT  TSAMP=T                                                           000130
CINT  PRINT%,' '                                                        000131
CINT  PRINT%,'ENTER +TIME ADVANCED, -TIME DELAY, OR 0.0'                000132
CINT  PRINT%,'SIGN OR VALUE SELECTS ADV, DELAY, OR STAND  '             000133
CINT  PRINT%,'Z-, W-, OR W'-TRANSFORM '                                 000134
CINT  PRINT%,'DELTA (SEC)='                                             000135
CINT  READ(5,%)AM                                                       000136
CINT  DELTA=AM                                                          000137
CINT  PRINT%,' '                                                        000138
CINT  IF(AM .LT. T) GO TO 2050                                          000139
CINT  PRINT%,'THE DELTA TIME IS GREATER THAN THE SAMPLING TIME'         000140
CINT  PRINT%,'DELTA (SEC)=',AM,'     TSAMP (SEC)=',T                    000141
CINT  PRINT%,'TRY AGAIN AND INPUT THESE VALUES'                         000142
CINT  PRINT%,' '                                                        000143
CINT  GO TO 2060                                                        000144
 2050 CONTINUE                                                          000145
C                                                                       000146
CINT  PRINT%,' '                                                        000147
 2090 CONTINUE                                                          000148
CINT  PRINT%,'D/A COUPLER OPTIONS AVAILABLE:'                           000149
CINT  PRINT%,'NONE ------------ NON'                                    000150
CINT  PRINT%,'ZERO-ORDER-HOLD -- ZOH'                                   000151
CINT  PRINT%,'1ST-ORDER-HOLD --- 1ST'                                   000152
CINT  PRINT%,'2ND-ORDER-HOLD --- 2ND'                                   000153
CINT  PRINT%,'SLEWER HOLD ------ SLE'                                   000154
CINT  PRINT%,'HOLD OPTION = '                                           000155
CINT  READ(5,2071) CPLR                                                 000156
 2071 FORMAT(A3)                                                        000157
CINT  NH=1                                                              000158
CINT  IF(CPLR .EQ. 3HNON) NH=0                                          000159
CINT  IF(CPLR .EQ. 3HNON .OR. CPLR .EQ. 3H1ST .OR. CPLR .EQ.            000160
CINT *3H2ND .OR. CPLR .EQ. 3HSLE .OR. CPLR .EQ. 3HZOH) GO TO 3030       000161
CINT  GO TO 2090                                                        000162
 3030 CONTINUE                                                          000163
CINT  PRINT%,' '                                                        000164
 3000 CONTINUE                                                          000165
```

```
CINT  PRINT%,'DISCRETE TRANSFORM OPTIONS AVAILABLE:'                    000166
CINT  PRINT%,'Z  TRANSFORM ----- Z'                                     000167
CINT  PRINT%,'U  TRANSFORM ----- U'                                     000168
CINT  PRINT%,'U' TRANSFORM ----- UP'                                    000169
CINT  PRINT%,'TRANSFORM OPTION = '                                      000170
CINT  READ(5,2181) TXFORM                                               000171
 2181 FORMAT(A2)                                                        000172
CINT  IF(TXFORM .EQ. 2HZ .OR. TXFORM .EQ. 2HU .OR. TXFORM .EQ.          000173
CINT % 2HUP) GO TO 3040                                                 000174
CINT  GO TO 3000                                                        000175
 3040 CONTINUE                                                          000176
CINT  PRINT%,' '                                                        000177
CINT  PRINT%,'TSAMP (SEC)=',T,'   DELTA (SEC)=',AM                      000178
CINT  WRITE(7,3151) CPLR,TXFORM                                         000179
 3151 FORMAT(1H ,'HOLD OPTION = ',A3,3X,'TRANSFORM OPTION = ',A2)       000180
CINT  PRINT%,' '                                                        000181
 3020 CONTINUE                                                          000182
CINT  PRINT%,'VARIABLE VALUES CORRECT (YES/NO)'                         000183
CINT  READ(5,3010) VALUE                                                000184
 3010 FORMAT(A2)                                                        000185
CINT  IF( VALUE .EQ. 2HYE .OR. VALUE .EQ. 2HNO) GO TO 3070              000186
CINT  GO TO 3020                                                        000187
 3070 CONTINUE                                                          000188
CINT  IF(VALUE .EQ. 2HNO) GO TO 2060                                    000189
CINT  PRINT%,' '                                                        000190
CINT  PRINT%,'ENTER NUMBER OF ZEROS AND POLES:'                         000191
CINT  READ(5,%) NRTN,M                                                  000192
CINT  PRINT%,'ENTER TRANSFER FUNCTION GAIN:'                            000193
CINT  READ(5,%) A0                                                      000194
CINT  IF(NRTN .EQ. 0) GO TO 4000                                        000195
CINT  PRINT%,'ENTER EACH ZERO --- RE,IM'                                000196
CINT  N2=2%NRTN                                                         000197
CINT  DO 4010 I=1,N2,2                                                  000198
CINT  II=(I+1)/2                                                        000199
CINT  PRINT%,'ZERO(',II,')= '                                           000200
CINT  READ(5,%) RN(I),RN(I+1)                                           000201
 4010 CONTINUE                                                          000202
 4000 CONTINUE                                                          000203
CINT  PRINT%,' '                                                        000204
CINT  IF(M .EQ. 0) GO TO 4020                                           000205
CINT  PRINT%,'ENTER EACH POLE -- RE,IM'                                 000206
CINT  M2=2%M                                                            000207
CINT  DO 4030 I=1,M2,2                                                  000208
CINT  II=(I+1)/2                                                        000209
CINT  PRINT%,'POLE(',II,')= '                                           000210
CINT  READ(5,%) RD(I),RD(I+1)                                           000211
 4030 CONTINUE                                                          000212
CINT  GO TO 4040                                                        000213
 4020 CONTINUE                                                          000214
CINT  PRINT%,'DENOMINATOR ORDER IS ZERO  -- EXIT PROGRAM'               000215
CINT  STOP                                                              000216
 4040 CONTINUE                                                          000217
CINT                                                                    000218
CINT  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  000219
CINT  %%%%% END READ SECTION FOR INTERACTIVE OPERATION %%%%%            000220
```

```
CINT  ***********************************************************      000221
C                                                                       000222
C    INPUT READ SECTION FOR BATCH MODE OPERATION                        000223
C                                                                       000224
      READ(5,*)NRTN,M,T,A0,AM                                           000225
      READ(5,2070) CPLR                                                 000226
 2070 FORMAT(A3)                                                        000227
      NH=1                                                              000228
      IF(CPLR .EQ. 3HNON) NH=0                                          000229
      READ(5,2080) TXFORM                                               000230
 2080 FORMAT(A2)                                                        000231
C                                                                       000232
C                                                                       000233
      IF(NRTN .EQ. 0) GO TO 2081                                        000234
      M2 = 2 * NRTN                                                     000235
      READ(5,*)(RN(I),I=1,M2)                                           000236
 2081 M2=2*M                                                            000237
      READ(5,*)(RD(I),I=1,M2)                                           000238
C                                                                       000239
CCC   END INPUT READ SECTION FOR BATCH MODE                             000240
CCC                                                                     000241
CCC                                                                     000242
CCC   ADD EQUAL ZERO AND POLE IF NRTN=0.                                000243
CCC   PARTR CANCELS THIS ZERO/POLE                                      000244
CCC   COMBINATION.                                                      000245
C                                                                       000246
      IF(NRTN .GT. 0) GO TO 2082                                        000247
      NRTN=1                                                            000248
      M2=2*NRTN                                                         000249
      RN(1)=1.123456789                                                 000250
      RN(2)=0.0                                                         000251
      DO 2083 I=1,M2                                                    000252
      II=M2-I+1                                                         000253
 2083 RD(II+2)=RD(II)                                                   000254
      RD(1)=1.123456789                                                 000255
      RD(2)=0.0                                                         000256
      M=M+1                                                             000257
      M2=2*M                                                            000258
 2082 CONTINUE                                                          000259
C                                                                       000260
CCC   END EQUAL ZERO AND POLE SECTION                                   000261
CCC                                                                     000262
CCC                                                                     000263
CCC   INSERT COMPLEX CONJUGATE ROOT IF                                  000264
CCC   NOT INPUTTED BY USER.                                             000265
C                                                                       000266
      J=2                                                               000267
   19 IF (RN(J)) 20,21,20                                               000268
   20 RN(J+1) = RN(J-1)                                                 000269
      RN(J+2) = -RN(J)                                                  000270
      J = J+4                                                           000271
      GO TO 28                                                          000272
   21 J=J+2                                                             000273
   28 IF(J-M2)  19,19,30                                                000274
   30 CONTINUE                                                          000275
```

```
      J=2                                                              000276
   24 IF(RD(J)) 22,23,22                                               000277
   22 RD(J+1) = RD(J-1)                                                000278
      RD(J+2) = -RD(J)                                                 000279
      J = J+4                                                          000280
      GO TO 29                                                         000281
   23 J=J+2                                                            000282
   29  IF(J-M2) 24,24,31                                               000283
   31 CONTINUE                                                         000284
C                                                                      000285
CC    END COMPLEX CONJUGATE ROOT SECTION                               000286
CCC                                                                    000287
CCC                                                                    000288
CC    IF ZOH SELECTED, ADD ADDITIONAL POLE                             000289
CC    TO THE INPUT G(S) TRANSFER FUNCTION.                             000290
C                                                                      000291
      IF (NH -1 ) 500, 4, 500                                          000292
  500 CONTINUE                                                         000293
      NRTD=M                                                           000294
      GO TO 5                                                          000295
4     NRTD=M+1                                                         000296
      L=M+M+1                                                          000297
      RD(L)=0.0                                                        000298
      RD(L+1)= 0.0                                                     000299
5     CONTINUE                                                         000300
CC                                                                     000301
CC    END ZOH SECTION                                                  000302
CCC                                                                    000303
CCC                                                                    000304
CC    INSERT ADDTIONAL ZEROS AND POLES IN THE                          000305
CC    S-DOMAIN FOR THE REMAINING DATA HOLDS                            000306
C                                                                      000307
      IF(CPLR .EQ. 3H1ST) GO TO 5000                                   000308
      IF(CPLR .EQ. 3HSLE) GO TO 5100                                   000309
      IF(CPLR .EQ. 3H2ND) GO TO 5200                                   000310
      GO TO 5300                                                       000311
 5000 NRTD=M+2                                                         000312
      L=M+M+3                                                          000313
      RD(L)=0.0                                                        000314
      RD(L+1)=0.0                                                      000315
      L1=NRTN+NRTN+1                                                   000316
      NRTN=NRTN+1                                                      000317
      RN(L1)= -1.0/T                                                   000318
      RN(L1+1)=0.0                                                     000319
      GO TO 5300                                                       000320
 5100 NRTD=M+2                                                         000321
      L=M+M+3                                                          000322
      RD(L)=0.0                                                        000323
      RD(L+1)=0.0                                                      000324
      GO TO 5300                                                       000325
 5200 NRTD=M+3                                                         000326
      L=M+M+3                                                          000327
      RD(L)=0.0                                                        000328
      RD(L+1)=0.0                                                      000329
      RD(L+2)=0.0                                                      000330
```

```
      RD(L+3)=0.0                                                       000331
      L11=NRTN+NRTN+1                                                   000332
      NRTN=NRTN+2                                                       000333
      RN(L11)= -0.75/T                                                  000334
      RN(L11+1)= -(2.645751311)/(4.0*T)                                 000335
      RN(L11+2)= -0.75/T                                                000336
      RN(L11+3)= (2.645751311)/(4.0*T)                                  000337
 5300 CONTINUE                                                          000338
C                                                                       000339
C     END ADDITIONAL ZEROS AND POLES SECTION                            000340
C                                                                       000341
C                                                                       000342
C     OUTPUT WRITE SECTION FOR BASIC DEFINING                           000343
C     PARAMETERS.                                                       000344
C                                                                       000345
      WRITE(7,3052)                                                     000346
 3052 FORMAT(1H1,///)                                                   000347
      PRINT*,' '                                                        000348
      PRINT*,'TSAMP (SEC)=',T,'   DELTA (SEC)=',AM                      000349
      WRITE(7,3050) CPLR,TXFORM                                         000350
 3050 FORMAT(1H ,*HOLD OPTION = *,A3,3X,*TRANSFORM OPTION = *,A2)       000351
      WRITE(7,3051) A0                                                  000352
 3051 FORMAT(1H ,*GAIN = *,D17.10)                                      000353
      PRINT*,' '                                                        000354
      PRINT*,' '                                                        000355
C                                                                       000356
C     END OUTPUT WRITE SECTION                                          000357
C                                                                       000358
C                                                                       000359
C                                                                       000360
C-----------------------------------------------------------------------000361
*     CALL OVERLAY(7HDISCRET,22,1)                                      000362
*     CALL OVERLAY(7HDISCRET,22,2)                                      000363
C-----------------------------------------------------------------------000364
C                                                                       000365
C                                                                       000366
C                                                                       000367
C-----------------------------------------------------------------------000368
      CALL PARTFR                                                       000369
      CALL WPLN                                                         000370
C-----------------------------------------------------------------------000371
C                                                                       000372
C                                                                       000373
      END                                                               000374
C                                                                       000375
C                                                                       000376
C                                                                       000377
C-----------------------------------------------------------------------000378
CDECK PARTFR                                                            000379
*     OVERLAY(26,1)                                                     000380
*     PROGRAM PARTFR                                                    000381
C-----------------------------------------------------------------------000382
C                                                                       000383
C                                                                       000384
                                                                        000385
```

```
C
C-----------------------------------------------------------------------
      SUBROUTINE PARTFR
C-----------------------------------------------------------------------
C
C
      COMMON/ADUCL2/RN,RD,NRTN,NRTD,MM,UR,UI,BR,BI,T,NM,AM,A0,
     +IXFORM,CPLR
      COMMON/TOTL12/CLNPOLY(51),CLDPOLY(51),CLZERO(50,2),
     +CLPOLE(50,2),NCLZ,NCLP,CLK,CLNK,CLDK
C
C
      DOUBLE CROOT(2),RN(102),RD(102)
      DIMENSION NEGLCT(51),MULT(51),MM(51)
      DOUBLE D1,D2,ONE,ZFN,SCALEF,D,ZFD,COEFF
      DOUBLE    UN(2),UD(2),RF(6),A(262),PN(612),PD(306),P2(102)
      DOUBLE UR(51),UI(51),BR(51),BI(51),T,AM,A0
    5 FORMAT(3(1X RE=,D16.8,1X   IM=D16.8))
    7 FORMAT(3(1X RE=,D16.8,1X   IM=D16.8))
   20 FORMAT (///18H0PARTIAL FRACTIONS //16X,17HN U M E R A T O R,31X,
     1        21HD E N O M I N A T O R/ 10X,9HREAL PART,11X,
     2        16HIMAG. PART,20X,9HREAL PART,11X,10HIMAG. PART,10X,
     3        8HEXPONENT )
   21 FORMAT (1PD22.8,D20.8,13X,D17.8,D20.8,11X,I2 )
   24 FORMAT (1H1)
      COEFF = A0 * 1.D-4
      NRN=NRTN+NRTN
      NRD=NRTD+NRTD
      ONE=1.
C     WRITE(7,24)
C     WRITE(7,6)
    6 FORMAT(1X INPUT ROOTS--NUMERATOR=//)
C     WRITE(7,5)(RN(I),I=1,NRN)
C     WRITE(7,1)
    1 FORMAT(////1X INPUT ROOTS--DENOMINATOR=//)
C     WRITE(7,5)(RD(I),I=1,NRD)
      L=1
  300 DO 390 I=L,NRN,2
      II=I
      DO 390 J=1,NRD,2
      JJ=J
      D1 =DABS(RN(I)* .0001)
      D2 =DABS(RN(I+1)* .0001)
      IF(RN(I))510,511,510
  510 IF(RD(J)) 512, 511,512
  511 D1 =.00001
  512 IF(RN(I+1))513,514,513
  513 IF(RD(J+1))515,514,515
  514 D2=.00001
  515 IF(DABS(RN(I)-RD(J))-D1)516,516,390
  516 IF(DABS(RN(I+1)-RD(J+1))-D2)517,517,390
  517 CONTINUE
      NRN=NRN-2
      NRD=NRD-2
      DO 370 K=II,NRN
```

```
  370  RN(K)=RN(K+2)                                                    000441
       DO 380 K=JJ,NRD                                                  000442
  380  RD(K)=RD(K+2)                                                    000443
       L=I                                                              000444
       IF (NRN-L) 400,390,390                                           000445
  390  CONTINUE                                                         000446
C *                        FIND T                                       000447
  400  L=1                                                              000448
       N=NRD/2                                                          000449
       DO 410 I=1,N                                                     000450
       NEGLCT(I)=0                                                      000451
  410  MULT(I)=1                                                        000452
  420  DO 440 I=L,NRD,2                                                 000453
       M=I+2                                                            000454
       IF (M-NRD) 518, 518, 450                                         000455
  518  CONTINUE                                                         000456
       DO 435 J=M,NRD,2                                                 000457
       D1 =DABS(RD(I)* .0001)                                           000458
       D2 =DABS(RD(I+1)* .0001)                                         000459
       IF( RD(I+1)) 519,520,519                                         000460
  519  IF(RD(J)) 521,520,521                                            000461
  520  D1=.00001                                                        000462
  521  CONTINUE                                                         000463
       IF( RD(I+1)) 522,523,522                                         000464
  522  IF( RD(J+1)) 524,523,524                                         000465
  523  D2=.00001                                                        000466
  524  CONTINUE                                                         000467
       IF(DABS(RD(I)-RD(J))-D1)525,525,435                              000468
  525  CONTINUE                                                         000469
       IF(DABS(RD(I+1)-RD(J+1))-D2)526,526,435                          000470
  526  CONTINUE                                                         000471
       NRD=NRD-2                                                        000472
       N=(I+1)/2                                                        000473
       MULT(N)=MULT(N)+1                                                000474
       DO 430 K=J,NRD                                                   000475
       N=(K+1)/2                                                        000476
       MULT(N)=MULT(N+1)                                                000477
  430  RD(K)=RD(K+2)                                                    000478
       L=I                                                              000479
       IF (NRD-L-1) 450,450,420                                         000480
  435  CONTINUE                                                         000481
  440  CONTINUE                                                         000482
  450  CONTINUE                                                         000483
       WRITE(7,2)                                                       000484
    2 FORMAT(//////* ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERAT000485
      1OR*//)                                                           000486
       WRITE(7,7)(RN(I),I=1,NRN)                                        000487
       WRITE(7,3)                                                       000488
    3 FORMAT(////* ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINAT000489
      1OR*//)                                                           000490
       WRITE(7,7)(RD(I),I=1,NRD)                                        000491
       RN(NRN+1) = 1.E-4                                                000492
       NRN=NRN/2                                                        000493
       CALL GETPOL(RN,NRN,A,NA)                                         000494
       K=1                                                              000495
```

```
      KKK=1                                                             000496
      DO 500 I=1,NRD,2                                                  000497
      M=I/2+1                                                           000498
      IFAC=0                                                            000499
      IF(NEGLCT(M))470,527,470                                          000500
  527 CONTINUE                                                          000501
      IF(RD(I+1)) 528,460,528                                           000502
  528 CONTINUE                                                          000503
      NEGLCT(M+1)=1                                                     000504
  460 NEGLCT(M)=1                                                       000505
      CALL FORM(RD,NRD,MULT,NEGLCT,P2,NP2)                              000506
  470 CALL CONVRT(A,NA,PN,NPN)                                          000507
      CALL CONVRT(P2,NP2,PD,NPD)                                        000508
      N=MULT(M)                                                         000509
      IF(RD(I+1)) 529,475,529                                           000510
  529 CONTINUE                                                          000511
      NRF=6                                                             000512
      RF(1)=1.                                                          000513
      RF(2)=0                                                           000514
      RF(3)=1.                                                          000515
      RF(4)=-RD(I)                                                      000516
      RF(5)=RD(I+1)                                                     000517
      RF(6)=0                                                           000518
      DO 474 J=1,N                                                      000519
      CALL MULTIP(PD,NPD,RF,NRF,A(103),L,3)                             000520
      CALL MLTPL(A(103),L,ONE,PD,NPD)                                   000521
  474 CONTINUE                                                          000522
  475 DO 490 J=1,N                                                      000523
       CROOT(1)=RD(I)                                                   000524
        CROOT(2)=RD(I+1)                                                000525
      IF(J-1)530,480,530                                                000526
  530 CONTINUE                                                          000527
      IFAC=IFAC+1                                                       000528
      CALL DIRIV(PN,NPN,PD,NPD)                                         000529
  480 CALL EVALUB(PN,NPN,CROOT,VN,ZFN)                                  000530
       VN(1)=VN(1)/FACT(IFAC)                                           000531
      VN(2)=VN(2)/FACT(IFAC)                                            000532
      CALL EVALUB(PD,NPD,CROOT,VD,ZFD)                                  000533
       SCALEF=ZFN/ZFD                                                   000534
      D=VD(1)                                                           000535
      IF(DABS(VD(1))-DABS(VN(1)))484,482,486                            000536
  482 IF(VD(1)) 486,531,486                                             000537
  531 CONTINUE                                                          000538
      D=VD(2)                                                           000539
      IF(DABS(VD(2))-DABS(VN(2)))532,533,533                            000540
  532 D=VN(2)                                                           000541
  533 CONTINUE                                                          000542
      GO TO 486                                                         000543
  484 D=VN(1)                                                           000544
  486 VD(1) =VD(1)/D                                                    000545
       VN(1)=(VN(1)/D)*SCALEF                                           000546
      VD(2)=VD(2)/D                                                     000547
       VN(2)=(VN(2)/D)*SCALEF                                           000548
       D=VD(1)**2+VD(2)**2                                              000549
      VR(KKK)=(VN(1)*VD(1)+VN(2)*VD(2))/D *COEFF                        000550
```

```
      VI(KKK)=(UD(1)*UN(2)-UN(1)*UD(2))/D*COEFF                          000551
      KKK=KKK+1                                                         000552
  490 K=K+2                                                             000553
      NEGLCT(M)=0                                                       000554
  500 CONTINUE                                                          000555
C     WRITE(7,24)                                                       000556
      WRITE(7,20)                                                       000557
      J=1                                                               000558
      JJJ=1                                                             000559
      DO 502 L=1,NRD,2                                                  000560
      N=L/2+1                                                           000561
      M=MULT(N)                                                         000562
      DO 501 I=1,M                                                      000563
      IX=M-I+1                                                          000564
      WRITE(7,21) VR(JJJ),VI(JJJ),RD(L),RD(L+1),IX                      000565
      II=(J+1)/2                                                        000566
      BR(JJJ)=RD(L)                                                     000567
      BI(JJJ)=RD(L+1)                                                   000568
      MM(II)=IX                                                         000569
      JJJ=JJJ+1                                                         000570
  501 J=J+2                                                             000571
  502 CONTINUE                                                          000572
      WRITE(7,24)                                                       000573
      NRTD=II                                                           000574
      RETURN                                                            000575
      END                                                               000576
C-----------------------------------------------------------------------000577
CDECK MULTIP                                                            000578
      SUBROUTINE MULTIP(C1,NT1,C2,NT2,C3,NT3,N)                         000579
C-----------------------------------------------------------------------000580
      DOUBLE    C1(1),C2(1),C3(1)                                       000581
      K=1-N                                                             000582
      DO 110 I=1,NT1,N                                                  000583
      DO 100 J=1,NT2,N                                                  000584
      K=K+N                                                             000585
      IF (N-3) 529,90,529                                               000586
  529 CONTINUE                                                          000587
      C3(K)=C1(I)*C2(J)                                                 000588
      C3(K+1)=C1(I+1)*C2(J+1)                                           000589
      GO TO 100                                                         000590
   90 C3(K)   =C1(I)*C2(J)-C1(I+1)*C2(J+1)                              000591
      C3(K+1)=C1(I)*C2(J+1)+C1(I+1)*C2(J)                               000592
      C3(K+2)=C1(I+2)+C2(J+2)                                           000593
  100 CONTINUE                                                          000594
      NT3=K+N-1                                                         000595
      IF(I-2) 110,530,530                                               000596
  530 CONTINUE                                                          000597
      CALL SIMPLE(C3,NT3,N)                                             000598
      K=NT3-N+1                                                         000599
  110 CONTINUE                                                          000600
      RETURN                                                            000601
      END                                                               000602
C-----------------------------------------------------------------------000603
CDECK MLTPL                                                             000604
      SUBROUTINE MLTPL(C,M,D,E,N)                                       000605
```

```
C-----------------------------------------------------------------------  000606
      DOUBLE C(1),E(1),D                                                  000607
      M=N                                                                 000608
      DO 1 I=1,N,2                                                        000609
      E(I+1)=C(I+1)                                                       000610
 1    E(I)=C(I)*D                                                         000611
      RETURN                                                              000612
      END                                                                 000613
C-----------------------------------------------------------------------  000614
CDECK GETPOL                                                              000615
      SUBROUTINE GETPOL(R,NR,S,MS)                                        000616
C-----------------------------------------------------------------------  000617
      DOUBLE    S(1),P(6,51),Q(102),R(1)                                  000618
      DIMENSION N(51)                                                     000619
  1   FORMAT (54H0CONSECUTIVE COMPLEX ROOTS  N O T  COMPLEX CONJUGATES.)  000620
      IF (NR) 6,5,6                                                       000621
  5   MS=2                                                                000622
       S(1)=R(1)                                                          000623
      S(2)=0                                                              000624
      GO TO 80                                                            000625
  6   FLAG =0                                                             000626
      DO 50 I=1,NR                                                        000627
      IF (FLAG) 49,7,49                                                   000628
  7   M=2*I                                                               000629
      IF (R(M)) 20,10,20                                                  000630
 10   P(1,I)=1.                                                           000631
      P(2,I)=1.                                                           000632
      P(3,I)=-R(M-1)                                                      000633
      P(4,I)=0                                                            000634
      N(I)=4                                                              000635
      GO TO 50                                                            000636
   20 IF(DABS(R(M-1)-R(M+1))-.000001)30,30,70                             000637
   30 IF(DABS(R(M)+R(M+2))-.000001)40,40,70                               000638
 40   P(1,I)=1.                                                           000639
      P(2,I)=2.                                                           000640
      P(3,I)=-2.*R(M-1)                                                   000641
      P(4,I)=1.                                                           000642
      P(5,I)=R(M-1)**2+R(M)**2                                            000643
      P(6,I)=0                                                            000644
      N(I)=6                                                              000645
      P(1,I+1)=1.                                                         000646
      P(2,I+1)=0                                                          000647
      N(I+1)=2                                                            000648
      FLAG=1.                                                             000649
      GO TO 50                                                            000650
 49   FLAG=0                                                              000651
 50   CONTINUE                                                            000652
      S(1)=R(2*NR+1)                                                      000653
      S(2)=0                                                              000654
      MS=2                                                                000655
      DO 60 I=1,NR                                                        000656
      CALL MULTIP(S,MS,P(1,I),N(I),Q,MQ,2)                                000657
      CALL MLTPL(Q,MQ,1.D0,S,MS)                                          000658
 60   CONTINUE                                                            000659
      GO TO 80                                                            000660
```

```
 70   WRITE(7,1)                                                        000661
      CALL DUMP                                                         000662
 80   RETURN                                                            000663
      END                                                               000664
C-----------------------------------------------------------------------000665
CDECK FORM                                                              000666
      SUBROUTINE FORM(R,NR,M,JUMP,P,NP)                                 000667
C-----------------------------------------------------------------------000668
      DOUBLE    P(1)                                                    000669
      DOUBLE R(1),TS(180)                                               000670
      DIMENSION M(1),JUMP(1)                                            000671
      KONJ=0                                                            000672
      K=1                                                               000673
      DO 3 I=1,NR,2                                                     000674
      IF (KONJ)2,531,2                                                  000675
  531 CONTINUE                                                          000676
      II=(I+1)/2                                                        000677
       IF (JUMP(II))2,500,2                                             000678
  500 CONTINUE                                                          000679
      N=M(II)                                                           000680
      IF(DABS(R(I)-R(I+2))-.00001)532,532,534                           000681
  532 IF(DABS(R(I+1)+R(I+3))-.00001)533,533,534                         000682
  533 KONJ=1                                                            000683
  534 CONTINUE                                                          000684
      IF((I+2)-NR) 535,535,536                                          000685
  536 KONJ=0                                                            000686
  535 CONTINUE                                                          000687
      DO 1 J=1,N                                                        000688
      TS(K)=R(I)                                                        000689
      TS(K+1)=R(I+1)                                                    000690
      K=K+2                                                             000691
      IF(KONJ) 537,1,537                                                000692
  537 CONTINUE                                                          000693
      TS(K)=R(I+2)                                                      000694
      TS(K+1)=R(I+3)                                                    000695
      K=K+2                                                             000696
 1    CONTINUE                                                          000697
      GO TO 3                                                           000698
 2    KONJ=0                                                            000699
 3    CONTINUE                                                          000700
      TS(K) = 1.D-8                                                     000701
      K=K/2                                                             000702
      CALL GETPOL(TS,K,P,NP)                                            000703
      RETURN                                                            000704
      END                                                               000705
C-----------------------------------------------------------------------000706
CDECK CONJRT                                                            000707
      SUBROUTINE CONJRT (P,NP,Q,NQ)                                     000708
C-----------------------------------------------------------------------000709
      DOUBLE P(1),Q(1)                                                  000710
      NQ=3*NP/2                                                         000711
      K=1                                                               000712
      DO 1 I=1,NP,2                                                     000713
      Q(K)=P(I)                                                         000714
      Q(K+1)=0                                                          000715
```

```
      Q(K+2)=P(I+1)                                                     000716
    1 K=K+3                                                             000717
      RETURN                                                            000718
      END                                                               000719
C-----------------------------------------------------------------------000720
CDECK DIRIV                                                             000721
      SUBROUTINE DIRIV(PN,NPN,PD,NPD)                                   000722
C-----------------------------------------------------------------------000723
      DOUBLE    PN(1),PD(1),A(306),B(306),C(306),ONE                    000724
      ONE = 1.0                                                         000725
      DO 1 I=1,NPN,3                                                    000726
      A(I)  =PN(I+2)*PN(I)                                              000727
      A(I+1)=PN(I+2)*PN(I+1)                                            000728
      A(I+2)=PN(I+2)-1.                                                 000729
    1 CONTINUE                                                          000730
      N=NPN                                                             000731
      CALL MULTIP(PD,NPD,A,N,B,M,3)                                     000732
      DO 2 I=1,NPD,3                                                    000733
      A(I)  =-PD(I+2)*PD(I)                                             000734
      A(I+1)=-PD(I+2)*PD(I+1)                                           000735
      A(I+2)=PD(I+2)-1.                                                 000736
    2 CONTINUE                                                          000737
      N=NPD                                                             000738
      CALL MULTIP(PN,NPN,A,N,C,K,3)                                     000739
      CALL ADD(B,M,C,K,PN,NPN,3)                                        000740
      CALL MULTIP(PD,NPD,PD,NPD,A,M,3)                                  000741
      CALL MLTPL(A,M,ONE,PD,NPD)                                        000742
      RETURN                                                            000743
      END                                                               000744
C-----------------------------------------------------------------------000745
CDECK EVALUB                                                            000746
      SUBROUTINE EVALUB (P,NP,R,U,ZF)                                   000747
C-----------------------------------------------------------------------000748
      DOUBLE    R(1),EXP,RP,CP,ZF                                       000749
      DOUBLE    P(1),U(2)                                               000750
      CALL ORDER3(P,NP,3)                                               000751
      ZF=1.                                                             000752
      P(NP+1)=0                                                         000753
      P(NP+2)=0                                                         000754
      P(NP+3)=0                                                         000755
      EXP=P(3)                                                          000756
       U(1)=P(1)                                                        000757
      U(2)=P(2)                                                         000758
      IF (EXP) 538,4,538                                                000759
  538 CONTINUE                                                          000760
      DO 3 I=1,NP,3                                                     000761
      IF(P(I+2)) 539,4, 539                                             000762
  539 CONTINUE                                                          000763
    1  RP=U(1)*R(1)-U(2)*R(2)                                           000764
       CP=U(1)*R(2)+U(2) *R(1)                                          000765
      IF(DABS(RP)-1.D20)11,540,540                                      000766
  540 CONTINUE                                                          000767
      ZF = ZF * 1.E10                                                   000768
      RP = RP/1.E10                                                     000769
      CP = CP / 1.E10                                                   000770
```

```
 11   CONTINUE                                                          000771
      EXP = EXP - 1.0                                                   000772
      IF(EXP- P(I+5)) 2,2,541                                           000773
 541  CONTINUE                                                          000774
       V(1)=RP                                                          000775
      V(2)=CP                                                           000776
      GO TO 1                                                           000777
    2  V(1)=RP+P(I+3)/ZF                                                 000778
       III=I+4                                                          000779
       V(2)=CP+P(III)/ZF                                                000780
 3    CONTINUE                                                          000781
 4    RETURN                                                            000782
      END                                                               000783
C-----------------------------------------------------------------------000784
CDECK FACT                                                              000785
      DOUBLE FUNCTION FACT(N)                                           000786
C-----------------------------------------------------------------------000787
      FACT=1.                                                           000788
      IF(N) 542,2,542                                                   000789
  542 CONTINUE                                                          000790
      P=N                                                               000791
    1 IF(P -1.) 543,2,543                                               000792
  543 CONTINUE                                                          000793
      FACT=FACT*P                                                       000794
      P=P-1.                                                            000795
      GO TO 1                                                           000796
 2    RETURN                                                            000797
      END                                                               000798
C-----------------------------------------------------------------------000799
CDECK ADD                                                               000800
      SUBROUTINE ADD(C1,NT1,C2,NT2,C3,NT3,N)                            000801
C-----------------------------------------------------------------------000802
      DOUBLE C1(1),C2(1),C3(1)                                          000803
      DO 100 I=1,NT1                                                    000804
  100 C3(I)=C1(I)                                                       000805
      DO 110 I=1,NT2                                                    000806
      J=I+NT1                                                           000807
  110 C3(J)=C2(I)                                                       000808
      NT3=NT1+NT2                                                       000809
      CALL SIMPLE(C3,NT3,N)                                             000810
      RETURN                                                            000811
      END                                                               000812
C-----------------------------------------------------------------------000813
CDECK SIMPLE                                                            000814
      SUBROUTINE SIMPLE(C,NT,N)                                         000815
C-----------------------------------------------------------------------000816
      DOUBLE C(2)                                                       000817
      M=NT-N                                                            000818
      DO 110 I=1,M,N                                                    000819
      JJ=I+N                                                            000820
      DO 110 J=JJ,NT,N                                                  000821
      K=M+I-1                                                           000822
      L=M+J-1                                                           000823
      IF(C(K)-C(L))110,100,110                                          000824
  100 C(I)=C(I)+C(J)                                                    000825
```

```
      C(J)=0                                                            000826
      IF(M-2)534,110,534                                                000827
  534 CONTINUE                                                          000828
      C(I+1)=C(I+1)+C(J+1)                                              000829
      C(J+1)=0                                                          000830
  110 CONTINUE                                                          000831
      II=1                                                              000832
  115 DO 140 I=II,NT,M                                                  000833
      IF(C(I))140,535,140                                               000834
  535 CONTINUE                                                          000835
      IF(M-2)536,120,536                                                000836
  536 CONTINUE                                                          000837
      IF(C(I+1))140,537,140                                             000838
  537 CONTINUE                                                          000839
  120 JJ=I+M                                                            000840
      DO 130 J=JJ,NT                                                    000841
      K=J-M                                                             000842
  130 C(K)=C(J)                                                         000843
      NT=NT-M                                                           000844
      II=I                                                              000845
      IF(I+M-1-NT)115,115,145                                           000846
  140 CONTINUE                                                          000847
  145 IF(NT)155,150,155                                                 000848
  150 NT=M                                                              000849
      C(1)=0                                                            000850
      C(2)=0                                                            000851
      C(3)=0                                                            000852
  155 RETURN                                                            000853
      END                                                               000854
C-----------------------------------------------------------------------000855
CDECK ORDER3                                                            000856
      SUBROUTINE ORDER3(P,N,K)                                          000857
C-----------------------------------------------------------------------000858
      DOUBLE T348R,T348I                                                000859
      DOUBLE P(1),TEMP                                                  000860
      M=N-K                                                             000861
      IF(M)5,5,1                                                        000862
    1 REVERS=1.                                                         000863
      DO 4 I=K,M,K                                                      000864
      J=I+K                                                             000865
      IF(P(I)-P(J))2,4,4                                                000866
    2 REVERS=0                                                          000867
      DO 3 L=1,K                                                        000868
      LJ=J-L+1                                                          000869
      LI=I-L+1                                                          000870
      TEMP=P(LJ)                                                        000871
      P(LJ)=P(LI)                                                       000872
      P(LI)=TEMP                                                        000873
    3 CONTINUE                                                          000874
    4 CONTINUE                                                          000875
      IF(REVERS)5,1,5                                                   000876
    5 RETURN                                                            000877
      END                                                               000878
C                                                                       000879
C                                                                       000880
```

```
C                                                                       000881
C                                                                       000882
C-----------------------------------------------------------------------000883
CDECK WPLN                                                              000884
%     OVERLAY(26,2)                                                     000885
%     PROGRAM WPLN                                                      000886
C-----------------------------------------------------------------------000887
C                                                                       000888
C                                                                       000889
C                                                                       000890
C-----------------------------------------------------------------------000891
      SUBROUTINE WPLN                                                   000892
C-----------------------------------------------------------------------000893
C                                                                       000894
C                                                                       000895
      DOUBLE XG,XSUM,T348R,T348I                                        000896
      DOUBLE VR(51),VI(51),BR(51),BI(51),AM,T,FR(51,5),FI(51,5),DR(51), 000897
     1DI(51),X,Y,X1,Y1,X2,Y2,X3,Y3,X4,Y4,X5,Y5,X6,Y6,XX,YY              000898
      DOUBLE CR(51,51),CI(51,51)                                        000899
      DOUBLE TIMESH                                                     000900
      DIMENSION MM(51)                                                  000901
      DIMENSION KK(51)                                                  000902
      DOUBLE AR(51),AI(51),RR(51),RI(51)                                000903
      DOUBLE RN(102),RD(102),A0                                         000904
      COMMON/ADUCZ/RN,RD,NRTN,NRTD,MM,VR,VI,BR,BI,T,NH,AM,A0,           000905
     +TXFORM,CPLR                                                       000906
      COMMON/TOTL12/CLNPOLY(51),CLDPOLY(51),CLZERO(50,2),               000907
     +CLPOLE(50,2),NCLZ,NCLP,CLK,CLNK,CLDK                              000908
C                                                                       000909
C                                                                       000910
  471 FORMAT(4D20.8)                                                    000911
  472 FORMAT(////)                                                      000912
  473 FORMAT(1X MM=,I5)                                                 000913
C                                                                       000914
      L1=NRTD                                                           000915
C                                                                       000916
C                                                                       000917
C   THE INPUT TIME SHIFT (AM = -, AM = 0, OR AM = +) DETERMINES         000918
C   THE TYPE OF Z-TRANSFORM CALCULATED (DELAYED FOR AM NEGATIVE;        000919
C   NORMAL FOR AM ZERO; AND ADVANCED FOR AM POSITIVE).                  000920
C   THE SHIFTING VARIABLE IS 'AMSH' AND THE ACTUAL TIME SHIFT           000921
C   IS STORED BACK IN 'AM'.                                             000922
C                                                                       000923
      TIMESH=AM                                                         000924
      IF(TIMESH) 1010,1020,1030                                         000925
 1010 AMSH=1.0                                                          000926
      AM=1+TIMESH/T                                                     000927
      GO TO 1000                                                        000928
 1020 AMSH=0.0                                                          000929
      AM=0.0                                                            000930
      GO TO 1000                                                        000931
 1030 AMSH=0.0                                                          000932
      AM=TIMESH/T                                                       000933
 1000 CONTINUE                                                          000934
C                                                                       000935
```

```
C                                                                       000936
      ND=51                                                             000937
      DO 300 I=1,51                                                     000938
  300 KK(I)=1                                                           000939
      L11=L1                                                            000940
      DO 50 I=1,51                                                      000941
      DO 50 J=1,6                                                       000942
      FR(I,J)=0.                                                        000943
      FI(I,J)=0.                                                        000944
   50 CONTINUE                                                          000945
      DO 51 I=1,51                                                      000946
      DO 51 J=1,51                                                      000947
      CR(I,J)=0.                                                        000948
   51 CI(I,J)=0.                                                        000949
      DO 100 I=1,L1                                                     000950
      K=MM(I)                                                           000951
      GO TO(20,30,40,53),K                                              000952
   20 CALL CDEXP(AM1T*BR(I),AM1T*BI(I),X,Y)                             000953
C     WRITE(7,12345) X,Y                                                000954
12345 FORMAT(' X = ',D38.30,5X,' Y = ',D38.30)                          000955
      CALL MULT(VR(I),VI(I),X,Y,X1,X2)                                  000956
C     WRITE(7,12345) X1,X2                                              000957
      CALL CDEXP(BR(I)*T,BI(I)*T,X,Y)                                   000958
C     WRITE(7,12345) X,Y                                                000959
      X=X+1.                                                            000960
      CALL DIVI(X1,X2,X,Y,FR(I,1),FI(I,1))                              000961
C     WRITE(7,12345) FR(I,1),FI(I,1)                                    000962
      FR(I,2)=-FR(I,1)                                                  000963
      FI(I,2)=-FI(I,1)                                                  000964
      GO TO 90                                                          000965
   30 CALL CDEXP(BR(I)*(AM+1)*T,BI(I)*(AM+1)*T,X,Y)                     000966
      CALL MULT(T*VR(I),T*VI(I),X,Y,X1,Y1)                              000967
      CALL CDEXP(BR(I)*T**(1+AM),BI(I)*T**(1+AM),X,Y)                   000968
      CALL MULT(AM1T*VR(I),AM1T*VI(I),X,Y,X2,Y2)                        000969
      X1=X1-X2                                                          000970
      Y1=Y1-Y2                                                          000971
      CALL CDEXP(BR(I)*AM1T,BI(I)*AM1T,X,Y)                             000972
      CALL MULT(VR(I)*AM1T,VI(I)*AM1T,X,Y,X2,Y2)                        000973
      CALL CDEXP(BR(I)*T,BI(I)*T,X,Y)                                   000974
      X=X+1.                                                            000975
      CALL MULT(X,Y,X,Y,X3,Y3)                                          000976
      CALL DIVI(X1+X2,Y1+Y2,X3,Y3,FR(I,1),FI(I,1))                      000977
      CALL DIVI(-2.*X1,-2.*Y1,X3,Y3,FR(I,2),FI(I,2))                    000978
      CALL DIVI(X1-X2,Y1-Y2,X3,Y3,FR(I,3),FI(I,3))                      000979
      GO TO 90                                                          000980
   40 CALL CDEXP(BR(I)*T**(AM+2),BI(I)*T**(AM+2),X,Y)                   000981
      CALL MULT(VR(I)*T**2,VI(I)*T**2,X,Y,X1,Y1)                        000982
      CALL CDEXP(BR(I)*T**(AM+1),BI(I)*T**(AM+1),X,Y)                   000983
      CALL MULT(VR(I)*(2*AM+1)*T**2/2.,VI(I)*(2*AM+1)*T**2/2,X,Y,X2,    000984
     1Y2)                                                               000985
      CALL CDEXP(BR(I)*AM1T,BI(I)*AM1T,X,Y)                             000986
      CALL MULT(VR(I)*T**2/2.*AM**2,VI(I)*T**2/2.*AM**2,X,Y,X3,Y3)      000987
      CALL CDEXP(BR(I)*T,BI(I)*T,X,Y)                                   000988
      CALL MULT(2.*X3,2.*Y3,X,Y,X4,Y4)                                  000989
      CALL MULT(X2,Y2,X,Y,X5,Y5)                                        000990
```

```
      CALL CDEXP(2*BR(I)*T,2*BI(I)*T,XX,YY)                                    000991
      CALL MULT(X3,Y3,XX,YY,X6,Y6)                                             000992
      X2=X2-X4                                                                 000993
      Y2=Y2-Y4                                                                 000994
      X1=X1-X5+X6                                                              000995
      Y1=Y1-Y5+Y6                                                              000996
      X=X+1                                                                    000997
      CALL MULT(X,Y,X,Y,X5,Y5)                                                 000998
      CALL MULT(X5,Y5,X,Y,X4,Y4)                                               000999
      CALL DIVI(X1+X2+X3,Y1+Y2+Y3,X4,Y4,FR(I,1),FI(I,1))                       001000
      CALL DIVI(X3-X2-3*X1,Y3-Y2-3*Y1,X4,Y4,FR(I,2),FI(I,2))                   001001
      CALL DIVI(3*X1-X2-X3,3*Y1-Y2-Y3,X4,Y4,FR(I,3),FI(I,3))                   001002
      CALL DIVI(X2-X1-X3,Y2-Y1-Y3,X4,Y4,FR(I,4),FI(I,4))                       001003
      GO TO 90                                                                 001004
   53 IF(DABS(BR(I)).LT.1.D-10.AND.DABS(BI(I)).LT.1.D-10)GO TO 54              001005
      WRITE(7,56)                                                              001006
   56 FORMAT(1X PROGRAM CANNOT HANDLE MULTIPLICITY OF 4 EXCEPT FOR K/S4)       001007
      CALL EXIT                                                                001008
   54 T348R=T**3/48.*VR(I)                                                     001009
      T348I=T**3/48.*VI(I)                                                     001010
      FR(I,1)=3.*T348R                                                         001011
      FI(I,1)=3.*T348I                                                         001012
      FR(I,2)=6.*(AM-1.)*T348R                                                 001013
      FI(I,2)=6.*(AM-1.)*T348I                                                 001014
      FR(I,3)=(6.*AM**2-12.*AM+2.)*T348R                                       001015
      FI(I,3)=(6.*AM**2-12.*AM+2.)*T348I                                       001016
      FR(I,4)=(4.*AM**3-12.*AM**2+6.*AM+2.)*T348R                              001017
      FI(I,4)=(4.*AM**3-12.*AM**2+6.*AM+2.)*T348I                              001018
      FR(I,5)=(-4.*AM**3+6.*AM**2-1.)*T348R                                    001019
      FI(I,5)=(-4.*AM**3+6.*AM**2-1.)*T348I                                    001020
   90 CALL CDEXP(BR(I)*T,BI(I)*T,X,Y)                                          001021
C     PRINT*,' '                                                               001022
C     PRINT*,' '                                                               001023
C     PRINT*,' '                                                               001024
C     WRITE(7,12345) X,Y                                                       001025
      CALL DIVI(1.-X,-Y,1+X,Y,DR(I),DI(I))                                     001026
C     PRINT*,' '                                                               001027
C     WRITE(7,12345) DR(I),DI(I)                                               001028
  100 CONTINUE                                                                 001029
      DO 204 I=1,L1                                                            001030
      DO 203 J=1,L1                                                            001031
      IF(I.EQ.J)GO TO 203                                                      001032
      A=DR(I)-DR(J)                                                            001033
      B=DI(I)-DI(J)                                                            001034
      IF(ABS(A).GT.1.E-13.OR.ABS(B).GT.1.E-13)GO TO 203                        001035
      IF(MM(I).LE.MM(J))KK(I)=0                                                001036
  203 CONTINUE                                                                 001037
  204 CONTINUE                                                                 001038
  474 FORMAT(//* DR,DI*/(2D20.8))                                              001039
  475 FORMAT(//* DR,VI*/(2D20.8))                                              001040
  476 FORMAT(10I5)                                                             001041
      DO 300 I=1,L1                                                            001042
      DO 210 J=1,5                                                             001043
      CR(I,J)=FR(I,J)                                                          001044
  210 CI(I,J)=FI(I,J)                                                          001045
```

```
      IE=MM(I)+1                                                        001046
      DO 250 J=1,L1                                                     001047
      IF(I.EQ.J)GO TO 250                                               001048
      IF(KK(J).EQ.0)GO TO 250                                           001049
      A=DR(I)-DR(J)                                                     001050
      B=DI(I)-DI(J)                                                     001051
      IF(ABS(A).LT.1.E-13.AND.ABS(B).LT.1.E-13)GO TO 230                001052
      K=MM(J)                                                           001053
  215 DO 228 IK=1,K                                                     001054
      DO 218 IM=1,IE                                                    001055
      IIE=IE-IM+1                                                       001056
      X=CR(I,IIE+1)                                                     001057
      CR(I,IIE+1)=CR(I,IIE+1)*DR(J)-CI(I,IIE+1)*DI(J)+CR(I,IIE)         001058
      CI(I,IIE+1)=CI(I,IIE+1)*DR(J)+          X*DI(J)+CI(I,IIE)         001059
C     WRITE(7,54321) CR(I,IIE+1),CI(I,IIE+1)                            001060
54321 FORMAT(' CR = ',D38.30,5X,'CI = ',D38.30)                         001061
  218 CONTINUE                                                          001062
      X=CR(I,1)                                                         001063
      CR(I,1)=CR(I,1)*DR(J)-CI(I,1)*DI(J)                               001064
      CI(I,1)=      X*DI(J)+CI(I,1)*DR(J)                               001065
C     PRINT*,' '                                                        001066
C     WRITE(7,54321) CR(I,1),CI(I,1)                                    001067
C     PRINT*,' '                                                        001068
  228 IE=IE+1                                                           001069
      GO TO 250                                                         001070
  230 IF(MM(J).LE.MM(I))GO TO 250                                       001071
      K=MM(J)-MM(I)                                                     001072
      GO TO 215                                                         001073
  250 CONTINUE                                                          001074
  300 CONTINUE                                                          001075
      JURIT=1                                                           001076
      DO 310 I=1,L1                                                     001077
      IF(KK(I).EQ.0)GO TO 310                                           001078
      JURIT=JURIT+MM(I)                                                 001079
  310 CONTINUE                                                          001080
      IF(L1.EQ.1)GO TO 317                                              001081
      DO 315 I=2,L1                                                     001082
      DO 315 J=1,JURIT                                                  001083
      CR(1,J)=CR(1,J)+CR(I,J)                                           001084
  315 CI(1,J)=CI(1,J)+CI(I,J)                                           001085
  317 CONTINUE                                                          001086
      DO 320 J=1,JURIT                                                  001087
      JIN=JURIT-J+1                                                     001088
      AR(JIN)=CR(1,J)                                                   001089
  320 AI(JIN)=CI(1,J)                                                   001090
      NN=JURIT-1                                                        001091
C                                                                       001092
C                                                                       001093
C   DEVIDE ALL NUMERATOR COEFFICIENTS BY THE                            001094
C   HIGHEST ORDER NUMERATOR COEFFICIENT.                                001095
C    NOTE THAT THIS OPERATION IS COMPLEX                                001096
C    NUMERS DIVIDED BY COMPLEX NUMBER.                                  001097
C                                                                       001098
C                                                                       001099
      XG=AR(1)                                                          001100
```

```
      Y6=AI(I)                                                          001101
      DO 488 I=1,JWRIT                                                  001102
      CALL DIVI(AR(I),AI(I),X6,Y6,X,Y)                                  001103
      AR(I)=X                                                           001104
  488 AI(I)=Y                                                           001105
C                                                                       001106
C                                                                       001107
C     WRITE(7,999) AR                                                   001108
C     WRITE(7,888) AI                                                   001109
C     PRINT*,' '                                                        001110
C     PRINT*,' END OF COEFF BEFORE CALL TO ROOTS'                       001111
C     PRINT*,' '                                                        001112
C     PRINT*,' '                                                        001113
      CALL ROOTS(AR,AI,NN,RR,RI)                                        001114
C     WRITE(7,999) AR                                                   001115
  999 FORMAT(' W NUM COEFF - REAL '/D38.30)                             001116
C     WRITE (7,888) AI                                                  001117
  888 FORMAT(' W NUM COEFF - IMAG '/D38.30)                             001118
  484 FORMAT(' AR'/(2D20.10))                                           001119
  328 CONTINUE                                                          001120
      IFLG=0                                                            001121
      IF(NN.EQ.0.AND.AMSH.NE.0.)GO TO 600                               001122
      A=1.                                                              001123
      IF(AMSH.NE.0)A=-1.                                                001124
  478 FORMAT(' ROOTS'/(2D20.8))                                         001125
      DO 330 I=1,NN                                                     001126
C                                                                       001127
      ISH=I                                                             001128
C                                                                       001129
      A1=RR(I)                                                          001130
      A2=RI(I)                                                          001131
      IF(ABS(A1-A).LE.1.E-11.AND.ABS(A2).LE.1.E-11)GO TO 340            001132
  330 CONTINUE                                                          001133
      DR(L1+1)=-A                                                       001134
      DI(L1+1)=0.                                                       001135
      L11=L1+1                                                          001136
      L1=L1+1                                                           001137
      MM(L11)=1                                                         001138
      KK(L11)=1                                                         001139
      IF(AMSH.EQ.0)X6=-X6                                               001140
      IF(AMSH.EQ.0)Y6=-Y6                                               001141
      GO TO 400                                                         001142
  340 DO 345 J=ISH,NN                                                   001143
      RR(J)=RR(J+1)                                                     001144
  345 RI(J)=RI(J+1)                                                     001145
      NN=NN-1                                                           001146
      IF(NN.EQ.1.AND.AMSH.NE.0)GO TO 348                                001147
      X6=-X6                                                            001148
      Y6=-Y6                                                            001149
  348 CONTINUE                                                          001150
  400 A=0.                                                              001151
  480 FORMAT(' DEN ROOTS'/(2D20.8))                                     001152
      IF(NN.EQ.0)A=1.                                                   001153
      DO 410 I=1,L1                                                     001154
C                                                                       001155
```

```
      ISH1=I                                                            001156
C                                                                       001157
      A1=DR(I)                                                          001158
      A2=DI(I)                                                          001159
      IF(ABS(A1-A).LE.1.E-12.AND.ABS(A2).LE.1.E-12.AND.KK(I).EQ.1)GO     001160
     1TO 420                                                            001161
  410 CONTINUE                                                          001162
      IF(NN.EQ.1)GO TO 415                                              001163
      NN=NN+1                                                           001164
      RR(NN)=-1.                                                        001165
      RI(NN)=0.                                                         001166
      GO TO 500                                                         001167
  415 NN=NN+1                                                           001168
      RR(NN)=0.                                                         001169
      RI(NN)=0.                                                         001170
      IFLG=2                                                            001171
      GO TO 500                                                         001172
C                                                                       001173
  420 I=ISH1                                                            001174
C                                                                       001175
      MM(I)=MM(I)-1                                                     001176
      IF(MM(I).EQ.0)KK(I)=0                                             001177
      IF(NN.EQ.1)IFLG=2                                                 001178
  470 CONTINUE                                                          001179
  500 CONTINUE                                                          001180
  600 CONTINUE                                                          001181
C                                                                       001182
CC                                                                      001183
C   SECTION TO ADD REQUIRED 1-E(-ST) * 2W/1+W TERMS TO THE W            001184
C   TRANSFER FUNCTION FOR THE SLEWER, 1ST, AND 2ND ORDER                001185
C   DATA HOLDS.                                                         001186
C                                                                       001187
C   THIS NEXT SUBSECTION HUNTS FOR (1+W) NUMERATOR ROOT AND             001188
C   CANCELS THEM INSTEAD OF ADDING (1+W) TO THE DENOMINATOR             001189
C   ROOTS.                                                              001190
C                                                                       001191
      LOOP=0                                                            001192
      GO TO 2950                                                        001193
 2900 LOOP=10                                                           001194
 2950 IF(CPLR .EQ. 3HZOH .OR. CPLR .EQ. 3HNON) GO TO 2600               001195
      A=-1.0                                                            001196
      DO 2330 I=1,NN                                                    001197
      ISH=I                                                             001198
      A1=RR(I)                                                          001199
      A2=RI(I)                                                          001200
      IF(ABS(A1-A).LE.1.E-11.AND.ABS(A2).LE.1.E-11) GO TO 2340          001201
 2330 CONTINUE                                                          001202
      DR(LI+1)=-A                                                       001203
      DI(LI+1)=0.0                                                      001204
      LI1=LI+1                                                          001205
      LI=LI+1                                                           001206
      MM(LI1)=1                                                         001207
      KK(LI1)=1                                                         001208
      GO TO 2400                                                        001209
 2340 DO 2345 J=ISH,NN                                                  001810
```

```
      RR(J)=RR(J+1)                                                     001211
 2345 RI(J)=RI(J+1)                                                     001212
      NN=NN-1                                                           001213
C                                                                       001214
C   THIS SUBSECTION HUNTS FOR (U=0.0) DENOMINATOR ROOTS AND             001215
C   CANCELS THEM INSTEAD OF ADDING (U=0.0) TO NUMERTAOR ROOTS.          001216
C                                                                       001217
 2400 CONTINUE                                                          001218
      A=0.0                                                             001219
      DO 2410 I=1,L1                                                    001220
      ISH1=I                                                            001221
      A1=DR(I)                                                          001222
      A2=DI(I)                                                          001223
      IF(ABS(A1-A).LE.1.E-12.AND.ABS(A2).LE.1.E-12.AND.KK(I).EQ.1)      001224
     * GO TO 2420                                                      001225
 2410 CONTINUE                                                          001226
      NN=NN+1                                                           001227
      RR(NN)=0.0                                                        001228
      RI(NN)=0.0                                                        001229
      GO TO 2500                                                        001230
 2420 I=ISH1                                                            001231
      MM(I)=MM(I)-1                                                     001232
      IF(MM(I) .EQ. 0) KK(I)=0                                          001233
 2500 CONTINUE                                                          001234
      IF(LOOP .GT. 1) GO TO 2600                                        001235
      IF(CPLR .EQ. 3H2ND) GO TO 2900                                    001236
 2600 CONTINUE                                                          001237
C                                                                       001238
C                                                                       001239
      I=0                                                               001240
  491 I=I+1                                                             001241
      IF(I.GT.L1)GO TO 496                                              001242
      IF(KK(I).EQ.1)GO TO 495                                           001243
      L1=L1-1                                                           001244
      DO 483 J=I,L1                                                     001245
      DR(J)=DR(J+1)                                                     001246
      DI(J)=DI(J+1)                                                     001247
      KK(J)=KK(J+1)                                                     001248
  483 MM(J)=MM(J+1)                                                     001249
      I=I-1                                                             001250
  495 GO TO 491                                                         001251
  496 CONTINUE                                                          001252
      AR(1)=-RR(1)                                                      001253
      AI(1)=-RI(1)                                                      001254
      AR(2)=1.                                                          001255
      AI(2)=0.                                                          001256
      IE=2                                                              001257
      IF(NN.EQ.1)GO TO 615                                              001258
      DO 610 I=2,NN                                                     001259
      DO 605 J=1,IE                                                     001260
      IIE=IE-J+1                                                        001261
      X=AR(IIE+1)                                                       001262
      AR(IIE+1)=AR(IIE)-RR(I)*AR(IIE+1)+RI(I)*AI(IIE+1)                 001263
  605 AI(IIE+1)=AI(IIE)-RI(I)*X-RR(I)*AI(IIE+1)                         001264
      IE=IE+1                                                           001265
```

```
      X=AR(1)                                                           001266
      AR(1)=-RR(I)*AR(1)+RI(I)*AI(1)                                    001267
  610 AI(1)=-RI(I)*X-RR(I)*AI(1)                                        001268
  615 CONTINUE                                                          001269
C                                                                       001270
C     NUMERATOR POLY GAIN FACTOR FOR ZERO ORDER HOLD (ZOH)              001271
C                                                                       001272
      IF(IFLG.EQ.2)X6=2.*X6                                             001273
      IF(IFLG.EQ.2)Y6=2.*Y6                                             001274
C                                                                       001275
C     NUMERATOR POLY GAIN FACTOR FOR FIRST ORDER HOLD (1ST)             001276
C                                                                       001277
      IF(CPLR .EQ. 3H1ST) X6=2.0*X6                                     001278
      IF(CPLR .EQ. 3H1ST) Y6=2.0*Y6                                     001279
C                                                                       001280
C     NUMERTOR POLY GAIN FACTOR FOR SECOND ORDER HOLD (2ND)             001281
C                                                                       001282
      IF(CPLR .EQ. 3H2ND) X6=4.0*X6                                     001283
      IF(CPLR .EQ. 3H2ND) Y6=4.0*Y6                                     001284
C                                                                       001285
C     NUMERTAOR GAIN FACTOR FOR SLEWER DATA HOLD (SLE)                  001286
C                                                                       001287
      IF(CPLR .EQ. 3HSLE) X6=(2.0/T)*X6                                 001288
      IF(CPLR .EQ. 3HSLE) Y6=(2.0/T)*Y6                                 001289
C                                                                       001290
C                                                                       001291
      DO 617 I=1,IE                                                     001292
      CALL MULT(AR(I),AI(I),X6,Y6,X,Y)                                  001293
      AR(I)=X                                                           001294
  617 AI(I)=Y                                                           001295
C                                                                       001296
C      TRANSFER LOGIC FOR Z OR W TRANSFORM                              001297
C                                                                       001298
      IF(TXFORM .EQ. 2HZ ) GO TO 3000                                   001299
      IF(TXFORM .EQ. 2HW ) GO TO 5000                                   001300
C                                                                       001301
C                                                                       001302
C     ****************  W' TRANSFORM SECTION  ****************          001303
C                                                                       001304
C                                                                       001305
C      THE W' TRANSFORM IS OBTAINED DIRECTLY FROM THE                   001306
C      W TRANSFORM BY USING THE FOLLOWING:                              001307
C                                                                       001308
C          W'=(ZEROS & POLES) = W=(ZEROS & POLES)*(2/T)                 001309
C          W'**2 (T/2)**2 *(A2) + W'**1 (T/2) * (A1) + A0 =             001310
C          W**2 * (A2) + W**1 * (A1) + A0                               001311
C                                                                       001312
C                                                                       001313
      DO 5200 I=1,IE                                                    001314
      IP=I-1                                                            001315
      AR(I)=((T/2)**IP)*AR(I)                                           001316
 5200 AI(I)=((T/2)**IP)*AI(I)                                           001317
C                                                                       001318
      DO 5300 I=1,MM                                                    001319
      RR(I)=(2/T)**RR(I)                                                001320
```

```
 5300 RI(I)=(2/T)*RI(I)                                                  001321
C                                                                        001322
 5000 CONTINUE                                                           001323
C                                                                        001324
      WRITE(7,619)                                                       001325
  619 FORMAT(///,10X,*COEFFICIENTS OF NUMERATOR IN W*/)                  001326
C                                                                        001327
      CLNK=AR(IE)                                                        001328
      X=AR(IE)                                                           001329
      DO 618 I=1,IE                                                      001330
      IEK=IE-I+1                                                         001331
      CLNPOLY(I)=AR(IEK)/X                                               001332
      IJ=I-1                                                             001333
      WRITE(7,621)CLNPOLY(I), IEK-1                                      001334
  621 FORMAT(1H ,10X,2(*,G18.10,5H)W** ,I3)                              001335
C     WRITE(7,620)AR(I),AI(I),IJ                                         001336
  620 FORMAT(5X,2(*,D38.30,*,*D38.30,5H)W** ,I3)                         001337
  618 CONTINUE                                                           001338
C                                                                        001339
      NCLZ=NN                                                            001340
      DO 3100 I=1,NN                                                     001341
      CLZERO(I,1)=RR(I)                                                  001342
 3100 CLZERO(I,2)=RI(I)                                                  001343
C                                                                        001344
C     WRITE(7,505)                                                       001345
  505 FORMAT(1H1)                                                        001346
      WRITE(7,506)                                                       001347
  506 FORMAT(///16X,*NUMERATOR ROOTS IN W*/)                             001348
      WRITE(7,516)(CLZERO(I,1),CLZERO(I,2),I=1,NN)                       001349
  516 FORMAT(5X,2(*,G18.10*,*G18.10*)*)                                  001350
C     WRITE(7,515)(RR(I),RI(I),I=1,NN)                                   001351
  515 FORMAT(5X,2(*,D38.30*,*D38.30*)*)                                  001352
C                                                                        001353
C      TRANSFER STATEMENT FOR Z  TRANSFORM                               001354
C                                                                        001355
 3000 CONTINUE                                                           001356
C                                                                        001357
      DO 650 I=1,51                                                      001358
      BR(I)=0.                                                           001359
  650 BI(I)=0.                                                           001360
      BR(1)=DR(1)                                                        001361
      BI(1)=DI(1)                                                        001362
      BR(2)=1.                                                           001363
      BI(2)=0.                                                           001364
      IIE=2                                                              001365
      DO 660 I=1,L1                                                      001366
      IF(I.EQ.1.AND.NN(I).EQ.1)GO TO 660                                 001367
      K=NN(I)                                                            001368
      IF(I.EQ.1)K=K-1                                                    001369
      DO 640 J=1,K                                                       001370
      DO 630 IJ=1,IIE                                                    001371
      IIJ=IIE-IJ+1                                                       001372
      XX=BR(IIJ+1)                                                       001373
      BR(IIJ+1)=BR(IIJ)+DR(I)*BR(IIJ+1)-DI(I)*BI(IIJ+1)                  001374
  630 BI(IIJ+1)=BI(IIJ)+DI(I)*XX+DR(I)*BI(IIJ+1)                         001376
```

```
      XX=BR(1)                                                          001376
      BR(1)=DR(I)*BR(1)-DI(I)*BI(1)                                     001377
      BI(1)=DI(I)*XX+DR(I)*BI(1)                                        001378
  640 IIE=IIE+1                                                         001379
  660 CONTINUE                                                          001380
C     WRITE(7,505)                                                      001381
      L11=IIE-1                                                         001382
      DO 549 I=1,L11                                                    001383
      DR(I)=-DR(I)                                                      001384
  549 DI(I)=-DI(I)                                                      001385
  550 FORMAT(///,14X,'DENOMINATOR ROOTS IN W'/)                         001386
C                                                                       001387
CC     TRANSFER LOGIC FOR Z OR W TRANSFORM                              001388
C                                                                       001389
      IF(TXFORM .EQ. 'HZ ') GO TO 3200                                  001390
      IF(TXFORM .EQ. 'HW ') GO TO 5100                                  001391
C                                                                       001392
CC   ****************  W' TRANSFORM SECTION  ****************           001393
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC 001394
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC 001395
CCCCC     THE W' TRANSFORM IS OBTAINED@ DIRECTLY FROM THE W              001396
CCCCC     TRANSFORM BY USING THE FOLLOWING:                              001397
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC 001398
CCCCC         W'-(ZEROS & POLES) = W -(ZEROS & POLES)*(2/T)              001399
CCCCC         W'**2 (T/2)**2 *(A2)  + W'**1 (T/2) * (A1) + A0 =          001400
CCCCC         W**2 * (A2) + W**1 * (A1) + A0                             001401
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC 001402
C                                                                       001403
      DO 5400 I=1,IIE                                                   001404
      IP=I-1                                                            001405
      BR(I)=((T/2)**IP)*BR(I)                                           001406
 5400 BI(I)=((T/2)**IP)*BI(I)                                           001407
C                                                                       001408
      DO 5600 I=1,L1                                                    001409
      DR(I)=(2/T)*DR(I)                                                 001410
 5600 DI(I)=(2/T)*DI(I)                                                 001411
C                                                                       001412
      CLDK=BR(IIE)                                                      001413
      MCLP=IIE-1                                                        001414
      CLK=CLMK/CLDK                                                     001415
      X=BR(IIE)                                                         001416
      DO 5700 I=1,IIE                                                   001417
 5700 BR(I)=BR(I)/X                                                     001418
      GO TO 5110                                                        001419
 5100 CONTINUE                                                          001420
      CLDK=1.0                                                          001421
      CLK=CLMK                                                          001422
      MCLP=IIE-1                                                        001423
 5110 CONTINUE                                                          001424
C                                                                       001425
C                                                                       001426
      WRITE(7,550)                                                      001427
      IEK=0                                                             001428
      DO 553 I=1,L1                                                     001429
      II=I                                                              001430
```

```
      K=MM(I)                                                           001431
      DO 554 J=1,K                                                      001432
      JK=II+J+IEK-1                                                     001433
      CLPOLE(JK,1)=DR(I)                                                001434
      CLPOLE(JK,2)=DI(I)                                                001435
      WRITE(7,516)CLPOLE(JK,1),CLPOLE(JK,2)                             001436
C     WRITE(7,3516) JK                                                  001437
 3516 FORMAT(1H1,X    ---------- JK = X,I6)                             001438
C     WRITE(7,515)DR(I),DI(I)                                           001439
  554 CONTINUE                                                          001440
      IEK=IEK+K-1                                                       001441
  553 CONTINUE                                                          001442
C     WRITE(7,505)                                                      001443
      WRITE(7,552)                                                      001444
  552 FORMAT(///,10X,XCOEFFICIENTS OF DENOMINATOR IN WX/)               001445
C                                                                       001446
      DO 551 I=1,IIE                                                    001447
      IJ=IIE-I                                                          001448
C                                                                       001449
      CLDPOLY(I)=BR(IJ+1)                                               001450
C                                                                       001451
      WRITE(7,621)CLDPOLY(I),IJ                                         001452
C     WRITE(7,620)BR(IJ+1),BI(IJ+1), J                                  001453
  551 CONTINUE                                                          001454
C                                                                       001455
C                                                                       001456
C     XXXXXXXXXXXXXXXX  Z - TRANSFORM SECTION XXXXXXXXXXXXXXXX         001457
C                                                                       001458
C     THIS SECTION IS SKIPPED FOR THE W OR W' OPTIONS.                   001459
C     THIS IS ACCOMPLISED WITH THE  GO TO 3400  STATEMENT.              001460
C                                                                       001461
C                                                                       001462
C      TRANSFER LOGIC FOR W OR W' TRANSFORM                             001463
      GO TO 3400                                                        001464
C                                                                       001465
C      TRANSFER STATEMENT FOR Z TRANSFORM                               001466
 3200 CONTINUE                                                          001467
C                                                                       001468
C                                                                       001469
      XG=0.                                                             001470
      DO 665 I=1,IE                                                     001471
  665 XG=XG+AR(I)                                                       001472
      B=XG                                                              001473
      IF(ABS(B).GT.1.E-7)GO TO 669                                      001474
      XG=0.                                                             001475
      DO 664 K=1,IE                                                     001476
      J=K-1                                                             001477
      I=IE-K                                                            001478
      IF(J.EQ.0)XG=XG+AR(K)*I                                           001479
      IF(J.EQ.0)GO TO 664                                               001480
      IF(I.EQ.0)XG=XG-AR(K)*J                                           001481
      IF(I.EQ.0)GO TO 664                                               001482
      XG=XG+AR(K)*(-J+I)                                                001483
  664 CONTINUE                                                          001484
      XSUM=0.                                                           001485
```

```
  669 DO 668 I=1,IIE                                                    001486
  668 XSUM=XSUM+BR(I)                                                   001487
      B=XSUM                                                            001488
      IF(ABS(B).GT.1.E-7)GO TO 662                                      001489
      XSUM=0.                                                           001490
      DO 663 K=1,IIE                                                    001491
      J=K-1                                                             001492
      I=IIE-K                                                           001493
      IF(J.EQ.0)XSUM=XSUM+BR(K)*I                                       001494
      IF(J.EQ.0)GO TO 663                                               001495
      IF(I.EQ.0)XSUM=XSUM-BR(K)+J                                       001496
      IF(I.EQ.0)GO TO 663                                               001497
      XSUM=XSUM+BR(K)*(-J+I)                                            001498
  663 CONTINUE                                                          001499
      IF(XSUM.EQ.0)WRITE(7,661)                                         001500
  662 XG=XG/XSUM                                                        001501
  661 FORMAT(' ERROR IN GAIN CALCULATION')                              001502
C     WRITE(7,505)                                                      001503
      DO 700 I=1,IE                                                     001504
      CR(1,I)=AR(I)                                                     001505
  700 CI(1,I)=AI(I)                                                     001506
      DO 705 I=1,IIE                                                    001507
      CR(2,I)=BR(I)                                                     001508
  705 CI(2,I)=BI(I)                                                     001509
      NK=NN                                                             001510
      DO 710 I=1,NN                                                     001511
  709 CONTINUE                                                          001512
      X=(1.-RR(I))**2+RI(I)**2                                          001513
      IF(DABS(X).GT.1.D-12)GO TO 707                                    001514
      NK=NK-1                                                           001515
      IF(I.GT.NK)GO TO 710                                              001516
      DO 708 J2=I,NK                                                    001517
      RR(J2)=RR(J2+1)                                                   001518
  708 RI(J2)=RI(J2+1)                                                   001519
      GO TO 709                                                         001520
  707 RR(I)=(1.-RR(I)**2-RI(I)**2)/X                                    001521
      RI(I)=2.*RI(I)/X                                                  001522
  710 CONTINUE                                                          001523
  805 CONTINUE                                                          001524
      NN=NK                                                             001525
      LK=L1                                                             001526
      DO 720 I=1,L1                                                     001527
  719 CONTINUE                                                          001528
      X=(1.-DR(I))**2+DI(I)**2                                          001529
      IF(DABS(X).GT.1.D-12)GO TO 718                                    001530
      LK=LK-1                                                           001531
      IF(I.GT.LK)GO TO 720                                              001532
      DO 717 J2=I,LK                                                    001533
      MM(J2)=MM(J2+1)                                                   001534
      KK(J2)=KK(J2+1)                                                   001535
      DR(J2)=DR(J2+1)                                                   001536
  717 DI(J2)=DI(J2+1)                                                   001537
      GO TO 719                                                         001538
  718 DR(I)=(1.-DR(I)**2-DI(I)**2)/X                                    001539
      DI(I)= 2.*DI(I)/X                                                 001540
```

```
  720 CONTINUE                                                          001541
      L1=LK                                                             001542
      WRITE(7,560)                                                      001543
      IF(IE.EQ.IIE)GO TO 570                                            001544
      IF(IE.GT.IIE)GO TO 572                                            001545
      K=IIE-IE                                                          001546
      DO 573 J=1,K                                                      001547
      NN=NN+1                                                           001548
      RR(NN)=-1.                                                        001549
  573 RI(NN)=0.                                                         001550
      GO TO 570                                                         001551
  572 L1=L1+1                                                           001552
      DR(L1)=-1.                                                        001553
      DI(L1)=0.                                                         001554
      MM(L1)=IE-IIE                                                     001555
      KK(L1)=1                                                          001556
  570 CONTINUE                                                          001557
  560 FORMAT(///,16X,'NUMERATOR ROOTS IN Z'/)                           001558
      IF(NN.EQ.0)GO TO 564                                              001559
C                                                                       001560
C                                                                       001561
      NCLZ=NN                                                           001562
C                                                                       001563
C                                                                       001564
      DO 3500 I=1,NN                                                    001565
      CLZERO(I,1)=RR(I)                                                 001566
      CLZERO(I,2)=RI(I)                                                 001567
 3500 CONTINUE                                                          001568
C                                                                       001569
C                                                                       001570
      WRITE(7,516)(CLZERO(I,1),CLZERO(I,2),I=1,NN)                      001571
C     WRITE(7,515)(RR(I),RI(I),I=1,NN)                                  001572
C 564 WRITE(7,585)                                                      001573
  564 WRITE(7,562)                                                      001574
  562 FORMAT(///,14X,'DENOMINATOR ROOTS IN Z'/)                         001575
      IEK=0                                                             001576
      DO 575 I=1,L1                                                     001577
      II=I                                                              001578
      K2=MM(I)                                                          001579
      DO 574 J=1,K2                                                     001580
C                                                                       001581
      JK=II+J+IEK-1                                                     001582
      CLPOLE(JK,1)=DR(I)                                                001583
      CLPOLE(JK,2)=DI(I)                                                001584
C                                                                       001585
C                                                                       001586
      WRITE(7,516)CLPOLE(JK,1),CLPOLE(JK,2)                             001587
C     WRITE(7,3616) JK                                                  001588
C     WRITE(7,515)DR(I),DI(I)                                           001589
  574 CONTINUE                                                          001590
      IEK=IEK+K2-1                                                      001591
  575 CONTINUE                                                          001592
      CALL POLYCO(FR,FI,RR,RI,NN)                                       001593
      NN1=NN+1                                                          001594
      DO 853 I=1,NN1                                                    001595
```

```
      J=MM1-I+1                                                          001596
      RR(I)=FR(J,1)                                                      001597
  853 RI(I)=FI(J,1)                                                      001598
      WRITE(7,857)                                                       001599
  857 FORMAT(///,10X,'COEFFICIENTS OF NUMERATOR IN Z'/)                  001600
      DO 566 I=1,MM1                                                     001601
      RR(I)=RR(I)*XG                                                     001602
  566 RI(I)=RI(I)*XG                                                     001603
C                                                                        001604
      CLNK=RR(MM1)                                                       001605
      CLK=RR(MM1)                                                        001606
      X=RR(MM1)                                                          001607
C                                                                        001608
      DO 854 I=1,MM1                                                     001609
C                                                                        001610
      IEK=MM1-I+1                                                        001611
      CLNPOLY(I)=RR(IEK)/X                                               001612
C                                                                        001613
      N1=I-1                                                             001614
      WRITE(7,856)CLNPOLY(I),IEK-1                                       001615
  856 FORMAT(1H ,10X,'(',G18.10,' )',5H Z** ,I5)                         001616
C     WRITE(7,157) CLNK,CLK                                              001617
  157 FORMAT(1H,'CLNK = ',G28.10,'  CLK = ',G28.10)                      001618
C     WRITE(7,855)RR(I),RI(I),N1                                         001619
  855 FORMAT(' ('D38.30,', ' D38.30,' )',4H Z**,I5)                      001620
  854 CONTINUE                                                           001621
      K2=L1                                                              001622
      DO 581 I=1,K2                                                      001623
      IF(MM(I).EQ.1)GO TO 581                                            001624
      J2=MM(I)-1                                                         001625
      DO 580 J=1,J2                                                      001626
      L1=L1+1                                                            001627
      DR(L1)=DR(I)                                                       001628
  580 DI(L1)=DI(I)                                                       001629
  581 CONTINUE                                                           001630
      CALL POLYCO(FR,FI,DR,DI,L1)                                        001631
      LP1=L1 +1                                                          001632
      DO 843 I=1,LP1                                                     001633
      J=LP1-I+1                                                          001634
      DR(I)=FR(J,1)                                                      001635
  843 DI(I)=FI(J,1)                                                      001636
      WRITE(7,873)                                                       001637
  873 FORMAT(///,10X,'COEFFICIENTS OF DENOMINATOR IN Z'/)                001638
C                                                                        001639
      NCLP=LP1-1                                                         001640
      CLDK=1.0                                                           001641
C                                                                        001642
      DO 848 I=1,LP1                                                     001643
C                                                                        001644
C                                                                        001645
      IEK=LP1-I+1                                                        001646
      CLDPOLY(I)=DR(IEK)                                                 001647
C                                                                        001648
C                                                                        001649
      I1=I-1                                                             001650
```

```
      WRITE(7,856)CLDPOLY(I),IEK-1                                      001651
C     WRITE(7,856)DR(I),DI(I),I1                                        001652
  848 CONTINUE                                                          001653
C                                                                       001654
C                                                                       001655
C      TRANSFER STATEMENT FOR U OR U' TRANSFORM                         001656
C                                                                       001657
 3400 CONTINUE                                                          001658
C                                                                       001659
C                                                                       001660
      WRITE(7,9977) CLK,CLNK,CLDK                                       001661
 9977 FORMAT(1H ,///,2X CLK=2,G18.10,/,2X CLNK=2,G18.10,/,2X CLDK=2,G18.10)001662
C     WRITE(7,5599) NCLP,NCLZ                                           001663
 5599 FORMAT(1H ,2HCLP =2,I5,5X,2HCLZ =2,I5)                            001664
C                                                                       001665
C                                                                       001666
      END                                                               001667
C-----------------------------------------------------------------------001668
CDECK ROOTS                                                             001669
      SUBROUTINE ROOTS(A,B,NN,RR,RI)                                    001670
C-----------------------------------------------------------------------001671
      DIMENSION A(1),B(1),RR(1),RI(1)                                   001672
      DOUBLE PRECISION A,B,RR,RI,C,D,C1,C2,D1,D2,X,Y,F,G,F1,F2,G1,G2,FH,001673
     1FK,FM                                                             001674
      L=1                                                               001675
      N=NN                                                              001676
      FM=1.                                                             001677
      NPLUS=N+1                                                         001678
      DO 265 I=1,NPLUS                                                  001679
      A1=A(I)                                                           001680
      B1=B(I)                                                           001681
      IF(ABS(A1).GT..00000001.OR.ABS(B1).GT..00000001)GO TO 266         001682
      DO 264 J=1,NPLUS                                                  001683
      A(J)=A(J+1)                                                       001684
  264 B(J)=B(J+1)                                                       001685
      N=N-1                                                             001686
      IF(N.EQ.0)RR(1)=0.                                                001687
      IF(N.EQ.0)RI(1)=0.                                                001688
      IF(N.EQ.0)RETURN                                                  001689
  265 CONTINUE                                                          001690
  266 CONTINUE                                                          001691
      NN=N                                                              001692
      DO 303 I=1,NN                                                     001693
      RR(I)=0.                                                          001694
  303 RI(I)=0.                                                          001695
  112 N1=N+1                                                            001696
      IF(N.EQ.1)GO TO 103                                               001697
      D1=DMAX1(DABS(A(N+1)),DABS(B(N+1)))                               001698
      IF(D1.EQ.0.)GO TO 111                                             001699
      D2=DMAX1(DABS(A),DABS(B))                                         001700
      FK=N                                                              001701
      FM=D2**(1.  /FK)/D1**(1.  /FK)                                    001702
      K=N                                                               001703
      DO 3 I=1,N                                                        001704
      A(I)=A(I)/(D1*FM**K)                                              001705
```

```
      B(I)=B(I)/(D1**FM**K)                                              001706
    3 K=K-1                                                              001707
      A(N1)=A(N1)/D1                                                     001708
      B(N1)=B(N1)/D1                                                     001709
    7 X=.9876532                                                         001710
      Y=.9654312                                                         001711
      LL=1                                                               001712
      IL=1                                                               001713
    8 C=A                                                                001714
      D=B                                                                001715
      C1=A                                                               001716
      C2=B                                                               001717
      DO 5 I=1,N                                                         001718
      F=X*C-Y*D+A(I+1)                                                   001719
      G=X*D+Y*C+B(I+1)                                                   001720
      IF(I.EQ.N)GO TO 5                                                  001721
      F1=X*C1-Y*C2+F                                                     001722
      F2=X*C2+Y*C1+G                                                     001723
      C=F                                                                001724
      D=G                                                                001725
      C1=F1                                                              001726
    5 C2=F2                                                              001727
      D=F1**2+F2**2                                                      001728
      FM=-(F*F1+G*F2)/D                                                  001729
      FK=-(G*F1-F*F2)/D                                                  001730
   12 X=X+FM                                                             001731
      Y=Y+FK                                                             001732
      IF((X**2+Y**2).EQ.((X+FM)**2+(Y+FK)**2))GO TO 22                   001733
      IF(X.EQ.0..OR.Y.EQ.0.)GO TO 22                                     001734
      A1=FM/X                                                            001735
      A2=FK/Y                                                            001736
      IF(ABS(A1  ).GT.1.E-4.OR. ABS(A2  ).GT.1.E-4)GO TO 21              001737
      GO TO(211,21),IL                                                   001738
  211 LL=194                                                             001739
      IL=2                                                               001740
   21 LL=LL+1                                                            001741
      IF(LL.GT.200)GO TO 22                                              001742
      GO TO 8                                                            001743
   22 RR(L)=X/FM                                                         001744
      RI(L)=Y/FM                                                         001745
      DO 100 I=1,N                                                       001746
      A(I+1)=X*A(I)-Y*B(I)+A(I+1)                                        001747
  100 B(I+1)=X*B(I)+Y*A(I)+B(I+1)                                        001748
  102 N=N-1                                                              001749
      L=L+1                                                              001750
      IF(N.GT.1)GO TO 7                                                  001751
  103 D=A**2+B**2                                                        001752
      RR(L)=(-A*A(2)-B*B(2))/(D*FM)                                      001753
      RI(L)=(-A*B(2)+B*A(2))/(D*FM)                                      001754
      DO 104 I=1,NM                                                      001755
      X=RR(I)**2                                                         001756
      Y=RI(I)**2                                                         001757
      D=X+Y                                                              001758
      IF(D.EQ.X)RI(I)=0.                                                 001759
      IF(D.EQ.Y)RR(I)=0.                                                 001760
```

```
  104 CONTINUE                                                          001761
      RETURN                                                            001762
  111 RR(L)=0.                                                          001763
      RI(L)=0.                                                          001764
      M=M-1                                                             001765
      L=L+1                                                             001766
      GO TO 112                                                         001767
      END                                                               001768
C-----------------------------------------------------------------------001769
CDECK CDEXP                                                             001770
      SUBROUTINE CDEXP(A,B,X,Y)                                         001771
C-----------------------------------------------------------------------001772
      DOUBLE A,B,X,Y                                                    001773
      X=DEXP(A)*DCOS(B)                                                 001774
      Y=DEXP(A)*DSIN(B)                                                 001775
      RETURN                                                            001776
      END                                                               001777
C-----------------------------------------------------------------------001778
CDECK MULT                                                              001779
      SUBROUTINE MULT(A,B,C,D,X,Y)                                      001780
C-----------------------------------------------------------------------001781
      DOUBLE A,B,X,Y,C,D                                                001782
      X=A*C-B*D                                                         001783
      Y=B*C+A*D                                                         001784
      RETURN                                                            001785
      END                                                               001786
C-----------------------------------------------------------------------001787
CDECK DIVI                                                              001788
      SUBROUTINE DIVI(A,B,C,D,X,Y)                                      001789
C-----------------------------------------------------------------------001790
      DOUBLE A,B,C,D,X,Y                                                001791
      X=(A*C+B*D)/(C**2+D**2)                                           001792
      Y=(B*C-A*D)/(C**2+D**2)                                           001793
      RETURN                                                            001794
      END                                                               001795
C-----------------------------------------------------------------------001796
CDECK POLYCO                                                            001797
      SUBROUTINE POLYCO(A,B,RR,RI,N)                                    001798
C-----------------------------------------------------------------------001799
      DIMENSION A(51,4),B(51,4),RR(1),RI(1)                             001800
      DOUBLE A,B,RR,RI                                                  001801
      NP1=N+1                                                           001802
      DO 1 I=2,NP1                                                      001803
      B(I,1)=0.                                                         001804
    1 A(I,1)=0.                                                         001805
      A(1,1)=1.                                                         001806
      B(1,1)=0.                                                         001807
      DO 3 I=1,N                                                        001808
       K=I+1                                                            001809
      DO 3 J=1,I                                                        001810
      A(K,1)=-A(K-1,1)*RR(I)+B(K-1,1)*RI(I)+A(K,1)                      001811
      B(K,1)=-B(K-1,1)*RR(I)-A(K-1,1 *RI(I)+B(K,1)                      001812
    3 K=K-1                                                             001813
      RETURN                                                            001814
      END                                                               001815
```

```
C     ********************************************************************** 000001
C     ********************************************************************** 000002
C     ********************************************************************** 000003
C     ************************  PROGRAM TXCONV  ************************** 000004
C     ********************************************************************** 000005
C     ********************************************************************** 000006
C     ********************************************************************** 000007
C                                                                            000008
C                                                                            000009
C-----------------------------------------------------------------------     000010
      OVERLAY(TXCONV,0,0)                                                    000011
      PROGRAM TXCONV(INPUT=100B,OUTPUT=100B,TAPE5=INPUT,TAPE7=OUTPUT)        000012
      CALL OVERLAY(6HTXCONV,23,0)                                            000013
      END                                                                    000014
C-----------------------------------------------------------------------     000015
C                                                                            000016
C                                                                            000017
C                                                                            000018
C                                                                            000019
C     ********************************************************************** 000020
C     *                                                                    * 000021
C     *       THE 'TXCONV' COMPUTER PROGRAM CALCULATES A LOW-RATE          * 000022
C     *  DISCRET TRANSFORM FROM A GIVEN HIGH-RATE DISCRETE TRANSFORM.      * 000023
C     *  THE INPUT TO 'TXCONV' IS A HIGH-RATE TRANSFER FUNCTION IN THE     * 000024
C     *  Z-, W-, OR W'-PLANE AND THE OUTPUT A LOW-RATE TRANSFER            * 000025
C     *  FUNCTION IN THE Z-, W-, OR W'-PLANE.                              * 000026
C     *                                                                    * 000027
C     *       TXCONV IS WRITTEN IN FORTRAN FOR THE CONTROL DATA            * 000028
C     *  CORPORATION (CDC) CYBER 175 SERIES COMPUTER.  THE PROGRAM CAN     * 000029
C     *  HANDLE POLE MULTIPLICITY UP TO THREE.  THE PROGRAM IS             * 000030
C     *  PRESENTLY SET-UP TO HANDLE SYSTEM ORDER UP TO 50TH.               * 000031
C     *  THE SYSTEM ORDER IS VARIABLE AND CAN BE EASILY CHANGED.  THE      * 000032
C     *  REQUIRED CHANGES TO THE DIMENSIONING STATEMENTS ARE OUTLINED      * 000033
C     *  BELOW.                                                            * 000034
C     *                                                                    * 000035
C     *       THERE IS NO RESTRICTION ON THE NUMBER OF SETS OF MULTIPLE    * 000036
C     *  POLES.  DOUBLE PRECISION ARITHEMATIC IS USED THROUGHOUT THE       * 000037
C     *  PROGRAM.                                                          * 000038
C     *                                                                    * 000039
C     *       THE FIVE AVAILABLE INPUT OPTIONS ARE OUTLINED BELOW:         * 000040
C     *                                                                    * 000041
C     *  'Z' OPTION -  INPUT HIGH-RATE ZEROS AND POLES IN Z-PLANE.         * 000042
C     *                OUTPUT LOW-RATE ZEROS AND POLES IN Z-PLANE.         * 000043
C     *                ALL CALCULATIONS IN THE W'-PLANE VIA THE            * 000044
C     *                POLYNOMIAL TRANSFORMATION METHOD, I.E.,             * 000045
C     *                HIGH-RATE Z-PLANE NUMERATOR AND DENOMINATOR         * 000046
C     *                POLYNOMIALS ARE FIRST TRANSFORMED TO                * 000047
C     *                W'-PLANE USING BILINEAR TRANSFORMATION.             * 000048
C     *                FOLLOWING MATHEMATICAL CALCULATIONS, LOW-RATE       * 000049
C     *                W'-PLANE TRANSFER FUNCTION IS TRANSFORMED           * 000050
C     *                BACK TO THE Z-PLANE.                                * 000051
C     *                                                                    * 000052
C     *  'W' OPTION -  INPUT HIGH-RATE ZEROS AND POLES IN W-PLANE.         * 000053
C     *                OUTPUT LOW-RATE ZEROS AND POLES IN W-PLANE.         * 000054
C     *                ALL CALCULATIONS IN THE W-PLANE.                    * 000055
```

```
C     *                                                                       * 000056
C     *  'WP' OPTION - INPUT HIGH-RATE ZEROS AND POLES IN W'-PLANE.           * 000057
C     *                OUTPUT LOW-RATE ZEROS AND POLES IN W'-PLANE.           * 000058
C     *                ALL CALCULATIONS IN THE W'-PLANE.                      * 000059
C     *                                                                       * 000060
C     *  'ZR' OPTION - THIS OPTION IS THE SAME AS THE Z OPTION                * 000061
C     *                EXCEPT THAT THE W'-PLANE HIGH-RATE POLES USED          * 000062
C     *                IN THE RESIDUE CALCULATIONS ARE OBTAINED BY            * 000063
C     *                DIRECT TRANSFORMATION OF THE INPUT Z-PLANE             * 000064
C     *                POLES.  THAT IS, THE HIGH-RATE W'-PLANE                * 000065
C     *                DENOMINATOR IS NOT ROOTED TO OBTAIN THESE              * 000066
C     *                POLES AS IS DONE IN THE Z OPTION.  THE ZR              * 000067
C     *                OPTION AVOIDS THE NUMERICAL ERRORS THAT MAY            * 000068
C     *                OCCUR WHEN ROOTING A POLYNOMIAL.                       * 000069
C     *                                                                       * 000070
C     *  'ZT' OPTION - INPUT HIGH-RATE ZEROS AND POLES IN THE Z-PLANE.        * 000071
C     *                OUTPUT LOW-RATE ZEROS AND POLES IN THE Z-PLANE.        * 000072
C     *                ALL CALCULATIONS IN THE Z-PLANE.                       * 000073
C     *                THIS OPTION LIMITED TO SIMPLE POLES,I.E., POLE         * 000074
C     *                MULTIPLICITY EQUAL TO ONE.                             * 000075
C     *                                                                       * 000076
C     *                                                                       * 000077
C     *  MOST INPUT DATA IS IN FREE-FORMAT WITH SEPARATORS RATHER THAN        * 000078
C     *  IN FIXED-SIZE FIELDS.  THE ONE EXCEPTION IS THE ALPHANUMERIC         * 000079
C     *  INPUT WHICH SELECTS THE TRANSFORMATION OPTION (Z, W, WP,             * 000080
C     *  ZR, OR ZT).  THE FREE-FORMAT INPUT DATA CONSISTS OF A STRING         * 000081
C     *  OF VALUES SEPARATED BY ONE OR MORE BLANKS, OR BY A COMMA OR          * 000082
C     *  SLASH, EITHER OF WHICH MAY BE PRECEEDED OR FOLLOWED BY ANY           * 000083
C     *  NUMBER OF BLANKS.  A LINE BOUNDARY, SUCH AS AN END OF RECORD         * 000084
C     *  OR END OF CARD, ALSO SERVES AS A VALUE SEPARATOR.                    * 000085
C     *                                                                       * 000086
C     *  THE REQUIRED INPUT DATA IS OUTLINED BELOW:                           * 000087
C     *                                                                       * 000088
C     *     DATA CARD          DATA ITEMS                 FORMAT             * 000089
C     *         1          GAIN,NZEROS,NPOLES         FREE-FORMAT           * 000090
C     *         2          TIN,TOUT                   FREE-FORMAT           * 000091
C     *         3          TRANSFORMATION OPTION          A3                * 000092
C     *         4          HIGH-RATE ZEROS            FREE-FORMAT           * 000093
C     *                    (REAL,IMAG)                                      * 000094
C     *       NTH          HIGH-RATE POLES            FREE-FORMAT           * 000095
C     *                    (REAL,IMAG)                                      * 000096
C     *                                                                       * 000097
C     *                                                                       * 000098
C     *          GAIN = HIGH-RATE TRANSFER FUNCTION GAIN                      * 000099
C     *          NZEROS = NUMBER OF ZEROS                                     * 000100
C     *          NPOLES = NUMBER OF POLES                                     * 000101
C     *          TIN = HIGH-RATE SAMPLING PERIOD (SEC)                        * 000102
C     *          TOUT = LOW-RATE SAMPLING PERIOD (SEC)                        * 000103
C     *          TRANSFORMATION OPTION = Z,W,WP,ZR, OR ZT                     * 000104
C     *          ZEROS(REAL,IMAG) = REAL AND IMAG PARTS OF ZEROS              * 000105
C     *          POLES(REAL,IMAG) = REAL AND IMAG PARTS OF POLES              * 000106
C     *                                                                       * 000107
C     *                                                                       * 000108
C     *  THE OUTPUT DATA FROM TXCONV IS DEVIDED INTO TWO MAIN SECTIONS.       * 000109
C     *  THE FIRST SECTION CONTAINS A LISTING OF THE INPUT DATA.              * 000110
```

```
C    *  THE SECOND SECTION GIVES THE LOW-RATE DISCRETE TRANSFER          * 000111
C    *  FUNCTION NUMERATOR AND DENOMINATOR POLYNOMIALS AND THEIR         * 000112
C    *  ROOTS.                                                           * 000113
C    *                                                                   * 000114
C    *                                                                   * 000115
C    ********************************************************************** 000116
C                                                                          000117
C                                                                          000118
C                                                                          000119
C    ********************************************************************** 000120
C    *                                                                   * 000121
C    *        **********  VARIABLES USED IN PROGRAM  **********         * 000122
C    *                                                                   * 000123
C    *  IADPOLE = FLAG TO ADD HIGH-RATE Z-PLANE POLE AT Z=0.0            * 000124
C    *            [Z = EXP(S*TINT)]                                      * 000125
C    *  KZERO = FLAG TO ADD (AP+WP) FACTOR TO W AND W' RESIDUE           * 000126
C    *          CALCULATIONS IN SUBROUTINES WMULT2 AND WMULT3.           * 000127
C    *  XAP,DXAP = HIGH-RATE ROOT IN W & W' TRANSFORMS                   * 000128
C    *             [I.E., (DXAP+WP), (DXAP-WP), AND (XAP+W)]             * 000129
C    *  DBCLK = TRANSFER FUNCTION GAIN                                   * 000130
C    *  NCLZDB = NUMBER OF ZEROS IN HIGH-RATE TRANSFORM                  * 000131
C    *  NCLPDB = NUMBER OF POLES IN HIGH-RATE TRANSFORM                  * 000132
C    *  TEXT = LOW-RATE SAMPLING INTERVAL (SEC).                         * 000133
C    *  TINT = HIGH-RATE SAMPLING INTERVAL (SEC).                        * 000134
C    *  XNT = RATIO OF SAMPLING INTERVALS (INTEGER OR NON-INTEGER)       * 000135
C    *  TXFORM = ALPHANUMERIC (A2) TRANSFORM OPTION INPUT:               * 000136
C    *           Z, ZR, ZT, W, OR WP.                                    * 000137
C    *  ZFORM = ALPHANUMERIC (A2) FLAG TO INDICATE                       * 000138
C    *          CONVERSION FROM Z-PLANE TO W'-PLANE                      * 000139
C    *          AND BACK AGAIN IN THE OUTPUT FROM                        * 000140
C    *          W'-PLANE TO Z-PLANE USING BILINEAR                       * 000141
C    *          TRANSFORMATION.                                          * 000142
C    *          IF  ZFORM=Z  ---  USE BILINEAR TRANSFORMATION            * 000143
C    *                            ON POLYS AND FACTOR TO GET             * 000144
C    *                            TRANSFORMED ZEROS AND POLES.           * 000145
C    *          IF ZFORM=ZR  ---  USE BILINEAR TRANSFORMATION            * 000146
C    *                            ON POLYS AND FACTOR TO GET             * 000147
C    *                            ONLY THE ZEROS AND USE                 * 000148
C    *                            STRAIGHT TRANSFORMATION ON             * 000149
C    *                            POLES TO OBTAIN TRANSFORMED            * 000150
C    *                            POLES.                                 * 000151
C    *  KEY = INTERGER THAT SPECIFIES THE PARTICULAR ROOT                * 000152
C    *        TRANSFORMATION DESIRED (E.G., Z-PLANE TO W'-PLANE,         * 000153
C    *        S-PLANE TO W-PLANE, ETC.).  SEE SUBROUTINE                 * 000154
C    *        'SZWROOT' FOR OPTIONS AVAILABLE.                           * 000155
C    *  T = SAMPLING PERIOD (SEC) TO BE USED IN THE ROOT                 * 000156
C    *      TRANSFORMATION SPECIFIED BY 'KEY'.                           * 000157
C    *  KTXFORM = INTERGER FLAG (KTXFORM=10) INDICATING THAT             * 000158
C    *            INPUT Z-PLANE POLYS AND ROOTS TRANSFORMED              * 000159
C    *            TO W'-PLANE FOR NUMERICAL CALCULATIONS.                * 000160
C    *            THIS FLAG SET EQUAL TO '10' WHEN TXFORM=Z              * 000161
C    *            OR TXFORM=ZR.                                          * 000162
C    *  NDPOLE = NUMBER OF DISTINCT POLES SENT TO SUBROUTINE             * 000163
C    *           'SZWROOT' FOR CONVERSION TO NEW COMPLEX                 * 000164
C    *           PLANE.                                                  * 000165
```

```
C   *  KFORM - INTERGER SPECIFYING FORMAT OF POLYNOMIALS.                    * 000166
C   *         2 - REAL COEFF, PWR                                           * 000167
C   *         3 - REAL COEFF, IMAG COEFF, PWR                               * 000168
C   *  ALPHA,SIGMA1,SIGMA2 - VARIABLES IN GENERAL BILINEAR                  * 000169
C   *                        TRANSFORMATION.                                * 000170
C   *                        Z,W,W'=ALPHA*[SIGMA1+Z,W,W']/                  * 000171
C   *                               [SIGMA2+Z,W,W']                         * 000172
C   *  KRES1Z - NUMBER OF ZEROS IN LOW-RATE TRANSFORM                       * 000173
C   *           ASSOCIATED WITH POLES WITH MULTIPLICITY=1.                  * 000174
C   *  KRES2Z - NUMBER OF ZEROS IN LOW-RATE TRANSFORM                       * 000175
C   *           ASSOCIATED WITH POLES WITH MULTIPLICITY=2.                  * 000176
C   *  KRES3Z - NUMBER OF ZEROS IN LOW-RATE TRANSFORM                       * 000177
C   *           ASSOCIATED WITH POLES WITH MULTIPLICITY=3.                  * 000178
C   *  KJRES2 - INTEGER FLAG FOR POLES WITH MULT=2.                         * 000179
C   *           THIS FLAG INVOKED WHEN TXFORM=Z OR                          * 000180
C   *           OR TXFORM=ZR (I.E., WHEN INPUT Z-PLANE                      * 000181
C   *           POLYS AND ROOTS TRANSFORMED TO W'-PLANE).                   * 000182
C   *           KJRES2=0  -  POLES OBTAINED THRU STRAIGHT                   * 000183
C   *                        TRANSFORMATION AND ZEROS THRU                  * 000184
C   *                        FACTORING NUMERATOR POLY.                      * 000185
C   *           KJRES2=10 -  POLES AND ZEROS OBTAINED THRU                  * 000186
C   *                        FACTORING NUM AND DEN POLYS.                   * 000187
C   *  KJRES3 - INTERGER FLAG FOR POLES WITH MULT=3.                        * 000188
C   *           SAME FUNCTION AS KJRES2.                                    * 000189
C   *  NPOLES - NUMBER OF DISTINCT POLES                                    * 000190
C   *  JRES1 - RESIDUE COUNTER FOR 1ST-ORDER POLES                          * 000191
C   *          (UPDATED IN SUBROUTINE ZMULT1 & WMULT1)                      * 000192
C   *  JRES2 - RESIDUE COUNTER FOR 2ND-ORDER POLES.                         * 000193
C   *         (UPDATED IN SUBROUTINE WMULT2)                                * 000194
C   *  JRES3 - RESIDUE COUNTER FOR 3RD-ORDER POLES.                         * 000195
C   *          (UPDATED IN SUBROUTINE WMULT3)                               * 000196
C   *  IPOLE - MASTER COUNTER FOR ALL RESIDUE CALCULATIONS.                 * 000197
C   *          (ORIGINATES IN MAIN PROGRAM)                                 * 000198
C   *  NM - NUMBER OF ELEMENTS IN MULPOLE                                   * 000199
C   *     - NUMBER OF DISTINCT POLES                                        * 000200
C   *  ND = 2 * (ORDER DBNPOLY + 1)                                         * 000201
C   *     - OR NUMBER OF OCCUPIED ELEMENTS IN DBNPOLY                       * 000202
C   *  ND = 2 * (ORDER DBDPOLY + 1)                                         * 000203
C   *     - OR NUMBER OF OCCUPIED ELEMENTS IN DBDPOLY                       * 000204
C   *  DBNPOLY - ARRAY CONTAINING HIGH-RATE NUMERATOR POLY:                 * 000205
C   *            [ (REAL COEFF) , (Z, W, OR W' PWR) ]                       * 000206
C   *  DBDPOLY - ARRAY CONTAINING HIGH-RATE DENOMINATOR POLY:               * 000207
C   *            [ (REAL COEFF) , (Z, W, OR W' PWR) ]                       * 000208
C   *  DBZERO - ARRAY CONTAINING HIGH-RATE ZEROS (REAL,IMAG)                * 000209
C   *  DBPOLE - ARRAY CONTAINING  HIGH-RATE POLES (REAL,IMAG)               * 000210
C   *  MULPOLE - ARRAY CONTAINING MULTIPLICITY OF EACH POLE                 * 000211
C   *  A,B,C3,C6 -  WORKING ARRAYS FOR INTERNAL CALCULATIONS                * 000212
C   *               THAT IS USED BY ALL ROUTINES.                           * 000213
C   *  RES1POL - ARRAY CONTAINING LOW-RATE TRANSFORM 1ST ORDER POLES        * 000214
C   *  RES2POL - ARRAY CONTAINING LOW-RATE TRANSFORM 2ND ORDER POLES        * 000215
C   *  RES3POL - ARRAY CONTAINING LOW-RATE TRANSFORM 3RD ORDER POLES        * 000216
C   *  RES1K - ARRAY CONTAINING NUMERATOR TERM FOR 1ST ORDER                * 000217
C   *          RESIDUES (FORMED IN SUBROUTINES ZMULT1 & WMULT1)             * 000218
C   *  RES2K1,RES2K2 - ARRAYS CONTAINING NUMERATOR TERMS FOR                * 000219
C   *                  2ND ORDER RESIDUES (SUBROUTINE WMULT2)               * 000220
```

```
C    *  RES3K1,RES3K2,RES3K3 - ARRAYS CONTAINING NUMERATOR TERMS FOR    * 000221
C    *                         3RD ORDER RESIDUES (SUBROUTINE WMULT3)   * 000222
C    *  RES1N - ARRAY CONTAINING LOW-RATE NUMERATOR POLY FOR POLES      * 000223
C    *          WITH MULTIPLICITY = 1 IN FORMAT: (REAL,IMAG,PWR).       * 000224
C    *          (FORMED IN SUBROUTINE RES1)                             * 000225
C    *  RES1D - ARRAY CONTAINING LOW-RATE DENOMINATOR POLY FOR POLES    * 000226
C    *          WITH MULTIPLICITY = 1 IN FORMAT: (REAL,IMAG,PWR).       * 000227
C    *          (FORMED IN SUBROUTINE RES1)                             * 000228
C    *  RES2N - ARRAY CONTAINING LOW-RATE NUMERATOR POLY FOR POLES      * 000229
C    *          WITH MULTIPLICITY = 2 IN FORMAT: (REAL,IMAG,PWR).       * 000230
C    *          (FORMED IN SUBROUTINE RES2)                             * 000231
C    *  RES2D - ARRAY CONTAINING LOW-RATE DENOMINATOR POLY FOR POLES    * 000232
C    *          WITH MULTIPLICITY = 2 IN FORMAT: (REAL,IMAG,PWR).       * 000233
C    *          (FORMED IN SUBROUTINE RES2)                             * 000234
C    *  RES3N - ARRAY CONTAINING LOW-RATE NUMERATOR POLY FOR POLES      * 000235
C    *          WITH MULTIPLICITY = 3 IN FORMAT: (REAL,IMAG,PWR).       * 000236
C    *          (FORMED IN SUBROUTINE RES3)                             * 000237
C    *  RES3D - ARRAY CONTAINING LOW-RATE DENOMINATOR POLY FOR POLES    * 000238
C    *          WITH MULTIPLICITY = 3 IN FORMAT: (REAL,IMAG,PWR).       * 000239
C    *          (FORMED IN SUBROUTINE RES3)                             * 000240
C    *  KRES1N - NUMBER OF OCCUPIED ELEMENTS IN ARRAY RES1N             * 000241
C    *  KRES1D - NUMBER OF OCCUPIED ELEMENTS IN ARRAY RES1D             * 000242
C    *  KRES2N - NUMBER OF OCCUPIED ELEMENTS IN ARRAY RES2N             * 000243
C    *  KRES2D - NUMBER OF OCCUPIED ELEMENTS IN ARRAY RES2D             * 000244
C    *  KRES3N - NUMBER OF OCCUPIED ELEMENTS IN ARRAY RES3N             * 000245
C    *  KRES3D - NUMBER OF OCCUPIED ELEMENTS IN ARRAY RES3D             * 000246
C    *  CLNPOLY - ARRAY CONTAINING NUMERATOR POLY COEFFICIENTS          * 000247
C    *            (ONLY) IN DECENDING ORDER (ZERO COEFF INCLUDED)       * 000248
C    *  CLDPOLY - ARRAY CONTAINING DENOMINATOR POLY COEFFICIENTS        * 000249
C    *            (ONLY) IN DESCENDING ORDER (ZERO COEFF INCLUDED)     * 000250
C    *  CLPOLE - ARRAY CONTAINING POLES (REAL,IMAG)                     * 000251
C    *  CLZERO - ARRAY CONTAINING ZEROS (REAL,IMAG)                     * 000252
C    *  NCLP - NUMBER OF POLES IN ARRAY CLPOLE                          * 000253
C    *  NCLZ - NUMBET OF ZEROS IN ARRAY CLZERO                          * 000254
C    *  CLNK - NUMERATOR POLY GAIN (ARRAY CLNPOLY)                      * 000255
C    *  CLDK - DENOMINATOR POLY GAIN (ARRAY CLDPOLY)                    * 000256
C    *  CLK - CLNK/CLDK                                                 * 000257
C    *     NOTE:  ARRAYS CLNPOLY, CLDPOLY, CLPOLE, AND CLZERO           * 000258
C    *            ARE USED FOR INITIAL INPUT OF DATA AND ALSO           * 000259
C    *            IN SUBROUTINES BILIN AND WZBILIN DURING               * 000260
C    *            EXECUTION OF THE MODES TXFORM=Z OR TXFORM=ZR.         * 000261
C    *  TOLR - TOLERANCE PERCENTAGE FOR REAL PART OF ROOT THAT          * 000262
C    *         IS USED IN THE ROOT CANCELLING SUBROUTINE CANROOT.       * 000263
C    *  TOLI - TOLERANCE PERCENTAGE FOR IMAG PART OF ROOT THAT          * 000264
C    *         IS USED IN THE ROOT CANCELLING SUBROUTINE CANROOT.       * 000265
C    *                                                                 * 000266
C    ******************************************************************* 000267
C                                                                         000268
C                                                                         000269
C                                                                         000270
C    ********************************************************************** 000271
C    *                                                                    * 000272
C    *           **********  ARRAY  DIMENSIONING  **********              * 000273
C    *                                                                    * 000274
C    *  NORDER - DENOMINATOR POLYNOMIAL ORDER                             * 000275
```

```
C    *                                                                    * 000276
C    *  CLNPOLY(NORDER+1)                                                 * 000277
C    *  CLDPOLY(NORDER+1)                                                 * 000278
C    *  CLZERO(NORDER,2)                                                  * 000279
C    *  CLPOLE(NORDER,2)                                                  * 000280
C    *  DBNPOLY(2*(NORDER+1))                                             * 000281
C    *  DBDPOLY(2*(NORDER+3))                                             * 000282
C    *  DBPOLE(NORDER+1,2)                                                * 000283
C    *  DBZERO(NORDER+1,2)                                                * 000284
C    *  RES1POL(NORDER+1,2)                                               * 000285
C    *  RES1K(NORDER+1,2)                                                 * 000286
C    *  RES2POL((NORDER+1)/2,2)                                           * 000287
C    *  RES2K1((NORDER+1)/2,2)                                            * 000288
C    *  RES2K2((NORDER+1)/2,2)                                            * 000289
C    *  RES3POL((NORDER+1)/3,2)                                           * 000290
C    *  RES3K1((NORDER+1)/3,2)                                            * 000291
C    *  RES3K2((NORDER+1)/3,2)                                            * 000292
C    *  RES3K3((NORDER+1)/3,2)                                            * 000293
C    *  RES1N(3*(NORDER+2))                                               * 000294
C    *  RES1D(3*(NORDER+2))                                               * 000295
C    *  RES2N(3*(NORDER+2))                                               * 000296
C    *  RES2D(3*(NORDER+2))                                               * 000297
C    *  RES3N(3*(NORDER+2))                                               * 000298
C    *  RES3D(3*(NORDER+2))                                               * 000299
C    *  A(2*(NORDER+3))                                                   * 000300
C    *  B(2*(NORDER+3))                                                   * 000301
C    *  C3(6*(NORDER+3))                                                  * 000302
C    *  C6(6*(NORDER+3))                                                  * 000303
C    *  MULPOLE(NORDER+1)                                                 * 000304
C    *                                                                    * 000305
C    ********************************************************************** 000306
C                                                                         000307
C                                                                         000308
C                                                                         000309
C                                                                         000310
C------------------------------------------------------------------------ 000311
CDECK MAIN                                                                000312
      OVERLAY(27,0)                                                       000313
      PROGRAM MAIN                                                        000314
C------------------------------------------------------------------------ 000315
C                                                                         000316
C                                                                         000317
C                                                                         000318
C------------------------------------------------------------------------ 000319
C     PROGRAM TXCONV(INPUT=100B,OUTPUT=100B,TAPE5=INPUT,TAPE7=OUTPUT)     000320
C------------------------------------------------------------------------ 000321
C                                                                         000322
C                                                                         000323
C                                                                         000324
      DOUBLE CLNPOLY(51),CLDPOLY(51),CLZERO(50,2),CLPOLE(50,2)            000325
      DOUBLE DBCLK,TINT,TEXT,XNT,T                                        000326
      DOUBLE ALPHA,SIGMA1,SIGMA2,CLK,CLNK,CLDK                            000327
      DOUBLE DBNPOLY(102),DBDPOLY(106),DBPOLE(51,2),DBZERO(51,2)          000328
      DOUBLE RES1N(156),RES2N(156),RES3N(156),RES1D(156),                 000329
     * RES2D(156),RES3D(156)                                             000330
```

```
      DOUBLE RES1POL(51,2),RES2POL(25,2),RES3POL(17,2)                  000331
      DOUBLE RES1K(51,2),RES2K1(25,2),RES2K2(25,2)                      000332
      DOUBLE RES3K1(17,2),RES3K2(17,2),RES3K3(17,2)                     000333
      DOUBLE A(106),B(106),C3(318),C6(318)                              000334
      DOUBLE AP(4),DXAP,TOLR,TOLI,PWR                                   000335
      DIMENSION MULPOLE(51)                                             000336
      COMMON/TXCONV9/CLNPOLY,CLDPOLY,CLZERO,CLPOLE,                     000337
     * NCLZ,NCLP,CLK,CLNK,CLDK                                          000338
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK         000339
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER                000340
      COMMON/ZEROS/KRES1Z,KRES2Z,KRES3Z                                 000341
      COMMON/TXCONV1/DBNPOLY,DBDPOLY,DBPOLE,DBZERO                      000342
      COMMON/TXCONV2/A,B,C3,C6                                          000343
      COMMON/TXCONV3/RES1K,RES1POL                                      000344
      COMMON/TXCONV4/RES2K1,RES2K2,RES2POL                              000345
      COMMON/TXCONV5/RES3K1,RES3K2,RES3K3,RES3POL                       000346
      COMMON/TXCONV6/RES1N,RES1D,KRES1N,KRES1D                          000347
      COMMON/TXCONV7/RES2N,RES2D,KRES2N,KRES2D                          000348
      COMMON/TXCONV8/RES3N,RES3D,KRES3N,KRES3D                          000349
C                                                                       000350
C                                                                       000351
C     ****************************************************************  000352
C     ******************  INPUT READ SECTION  ************************  000353
C     ****************************************************************  000354
C                                                                       000355
      WRITE(7,600)                                                      000356
  600 FORMAT(1H ,*INPUT DATA IN FOLLWING ORDER:*,/,* (1) CLK,*          000357
     1 * NCLZ, NCLP*,/,* (2) TINT, TEXT*,/,* (3) TXFORM*,/,             000358
     2 * (4) CLZERO (REAL,IMAG)*,/,* (5) CLPOLE (REAL,IMAG)*,/)         000359
      READ(5,*) CLK,NCLZ,NCLP                                           000360
      READ(5,*) TINT,TEXT                                               000361
      READ(5,601) TXFORM                                                000362
  601 FORMAT(A2)                                                        000363
      IF(NCLZ .EQ. 0) GO TO 602                                         000364
      READ(5,*) (CLZERO(I,1),CLZERO(I,2),I=1,NCLZ)                      000365
  602 READ(5,*) (CLPOLE(I,1),CLPOLE(I,2),I=1,NCLP)                      000366
C                                                                       000367
C                                                                       000368
C  INITIALIZE VARIABLES                                                 000369
      NORDER=50                                                         000370
      KRES1N=0                                                          000371
      KJRES2=0                                                          000372
      KJRES3=0                                                          000373
      KZERO=0                                                           000374
      IADPOLE=0                                                         000375
      NCLZDB=NCLZ                                                       000376
      NCLPDB=NCLP                                                       000377
      XNT=TEXT/TINT                                                     000378
      DBCLK=CLK                                                         000379
      KTXFORM=0                                                         000380
      ZFORM=2HBK                                                        000381
C                                                                       000382
C                                                                       000383
C  OBTAIN NUMERATOR POLY FROM ZEROS                                     000384
C                                                                       000385
```

```
      IF(NCLZ .EQ. 0) GO TO 27                                          000386
      DO 25 I=1,NCLZ                                                    000387
      A(I)=CLZERO(I,1)                                                  000388
   25 B(I)=CLZERO(I,2)                                                  000389
      CALL COMPOLY(CLNPOLY,C3,A,B,NCLZ)                                 000390
      GO TO 28                                                          000391
   27 CLNPOLY(1)=1.0                                                    000392
C                                                                       000393
C  OBTAIN DENOMINATOR POLY FROM POLES                                   000394
C                                                                       000395
   28 DO 26 I=1,NCLP                                                    000396
      A(I)=CLPOLE(I,1)                                                  000397
   26 B(I)=CLPOLE(I,2)                                                  000398
      CALL COMPOLY(CLDPOLY,C3,A,B,NCLP)                                 000399
C                                                                       000400
C                                                                       000401
C     **************************************************************    000402
C     ****************  WRITE SECTION FOR INPUT  *******************    000403
C     **************************************************************    000404
C                                                                       000405
      WRITE(7,8) XNT,TINT,TEXT                                          000406
    8 FORMAT(1H1,*RATIO =*,G15.8,4X,*TIN =*,G15.8,4X,*TOUT =*,G15.8)    000407
      WRITE(7,9) DBCLK,NCLZDB,NCLPDB                                    000408
    9 FORMAT(1H ,*GAIN =*,G15.8,5X,*NZEROS =*,I3,13X,*NPOLES =*,I3)     000409
      WRITE(7,701) TXFORM                                               000410
  701 FORMAT(1H ,///,11X,A2,*-PLANE  HIGH-RATE NUMERATOR*,/)            000411
      NZ=NCLZ+1                                                         000412
      DO 203 I=1,NZ                                                     000413
      K=NCLZ-I+1                                                        000414
  203 WRITE(7,702) CLNPOLY(I),TXFORM,K                                  000415
  702 FORMAT(1H ,10X,*(*,G18.10,* )*,A2,3H** ,I3)                       000416
      WRITE(7,703) TXFORM                                               000417
  703 FORMAT(1H ,///,10X,A2,*-PLANE  HIGH-RATE DENOMINATOR*,/)          000418
      NP=NCLP+1                                                         000419
      DO 204 I=1,NP                                                     000420
      K=NCLP-I+1                                                        000421
  204 WRITE(7,702) CLDPOLY(I),TXFORM,K                                  000422
      WRITE(7,704) TXFORM                                               000423
  704 FORMAT(1H ,///,14X,A2,*-PLANE  HIGH-RATE ZEROS*,/)                000424
      IF(NCLZ .EQ. 0) GO TO 603                                         000425
      WRITE(7,705) (CLZERO(I,1),CLZERO(I,2),I=1,NCLZ)                   000426
  705 FORMAT(1H ,5X,*(*,G18.10,*,*,G18.10,* )*)                         000427
  603 WRITE(7,706) TXFORM                                               000428
  706 FORMAT(1H ,///,14X, A2,*-PLANE  HIGH-RATE POLES*,/)               000429
      WRITE(7,705) (CLPOLE(I,1),CLPOLE(I,2),I=1,NCLP)                   000430
C                                                                       000431
C                                                                       000432
C    ***************************************************************    000433
C    ***************  Z-PLANE TO UP-PLANE TRANSFER  ****************    000434
C    ***************************************************************    000435
C                                                                       000436
      IF(TXFORM .NE. 2HZ .AND. TXFORM .NE. 2HZR) GO TO 8000             000437
C                                                                       000438
C   TRANSFORM INPUT Z-PLANE POLYS AND ROOTS TO W'-PLANE                 000439
C   USING BILINEAR TRANSFORMATION:                                      000440
```

```
C                                                                       000441
CC           Z = [(2/T)+UP]/[(2/T)-UP]                                  000442
CCC                                                                     000443
CCC          Z = ALPHA*[SIMA1+UP]/[SIMA2+UP]                            000444
C                                                                       000445
      NDPOLE=NCLP                                                       000446
      ZFORM=2HZ                                                         000447
      IF(TXFORM .EQ. 2HZR) ZFORM=2HZR                                   000448
      KEY=2                                                             000449
      T=TINT                                                            000450
      ALPHA= -1.0                                                       000451
      SIGMA1= 2.0/TINT                                                  000452
      SIGMA2= -2.0/TINT                                                 000453
C                                                                       000454
      CALL BILIN(ALPHA,SIGMA1,SIGMA2,T,KEY,ZFORM,NDPOLE)                000455
      NCLZDB=NCLZ                                                       000456
      NCLPDB=NCLP                                                       000457
      TXFORM=2HUP                                                       000458
      KTXFORM=10                                                        000459
C                                                                       000460
CC                                                                      000461
 8000 CONTINUE                                                          000462
C                                                                       000463
CCCC                                                                    000464
CCCC  ********************************************************************000465
CCCC  ******************  NUMERATOR POLY TRANSFER  ***********************000466
CCCCC ********************************************************************000467
CCCC                                                                    000468
CCCC  TRANSFER SINGLE PRECISION ARRAY CLNPOLY [FORMAT: REAL COEFFS]     000469
CCCC  INTO DOUBLE PRECISION ARRAY DBNPOLY [ REAL COEF,PWR]              000470
CCCC  WITH ALL ZERO COFFICIENTS IN CLNPOLY DELETED.                     000471
CCCCC        NN = 2 * (NCLZ+1)   ---- NUMBER DBNPOLY ELEMENTS          000472
CCCCC        NCLZ = NUMBER  ZEROS IN HIGH-RATE DISCRETE TRANSFORM       000473
C                                                                       000474
      J=-1                                                              000475
      K=NCLZ+1                                                          000476
      KK=NCLZ+1                                                         000477
      DO 50 I=1,K                                                       000478
C  COUNTER FOR DECENDING Z-PWR, W-PWR, OR W'-PWR                        000479
      KK=KK-1                                                           000480
C  CHECK FOR ZERO COEFFICIENT IN CLNPOLY                                000481
      IF(CLNPOLY(I)) 62,50,62                                           000482
C  TRANSFER  COEFFS & ADD Z-PWR, W-PWR, OR W'-PWR TO DBNPOLY            000483
   62 J=J+2                                                             000484
      DBNPOLY(J)=CLNPOLY(I)                                             000485
      DBNPOLY(J+1)=KK                                                   000486
   50 CONTINUE                                                          000487
C  SET  DBNPOLY ELEMENTS,I.E.,DBNPOLY(NN)                               000488
      NN=J+1                                                            000489
C  ORDER DBNPOLY  IN DECENDING POWERS OF Z, W, OR W'                    000490
      CALL ORDER3(DBNPOLY,NN,2)                                         000491
C                                                                       000492
CC                                                                      000493
C  ********************************************************************000494
C  ******************  DENOMINATOR POLY TRANSFER  *********************000495
```

```
C   ***************************************************************      000496
C                                                                        000497
C   TRANSFER SINGLE PRECISION ARRAY CLDPOLY [FORMAT: REAL COEFFS]        000498
C   INTO DOUBLE PRECISION ARRAY DBDPOLY [ REAL COEF,PWR]                 000499
C   WITH ALL ZERO COEFFICIENTS IN CLDPOLY DELETED.                       000500
C         ND = 2 * (NCLP+1)   ---- NUMBER DBDPOLY ELEMENTS               000501
C         NCLP = NUMBER  POLES IN HIGH-RATE DISCRETE TRANSFORM           000502
C                                                                        000503
      J=-1                                                               000504
      K=NCLP+1                                                           000505
      KK=NCLP+1                                                          000506
      DO 75 I=1,K                                                        000507
C  COUNTER FOR DECENDING POWERS OF Z, W, OR W'                           000508
      KK=KK-1                                                            000509
C  CHECK FOR ZERO COEFF IN CLDPOLY                                       000510
      IF(CLDPOLY(I)) 84,75,84                                            000511
C  TRANSFER COEFFS  & ADD Z-PWR, W-PWR, OR W'-PWR TO DBDPOLY             000512
   84 J=J+2                                                              000513
      DBDPOLY(J)=CLDPOLY(I)                                              000514
      DBDPOLY(J+1)=KK                                                    000515
   75 CONTINUE                                                           000516
C  SET  DBDPOLY ELEMENTS,I.E., DBDPOLY(ND)                               000517
      ND=J+1                                                             000518
C  ORDER DBDPOLY IN DECENDING POWERS OF Z, W, OR W'                      000519
      CALL ORDER3(DBDPOLY,ND,2)                                          000520
C                                                                        000521
C                                                                        000522
C   ***************************************************************      000523
C   ******************  POLE AND ZERO TRANSFER  *******************      000524
C   ***************************************************************      000525
C                                                                        000526
C   TRANSFER SINGLE PRECISION POLES AND ZEROS IN CLPOLE(50,2)            000527
C   AND CLZERO(50,2) INTO DOUBLE PRECISION ARRAYS DBPOLE(51,2)           000528
C   AND DBZERO(51,2).                                                    000529
C                                                                        000530
      DO 150 I=1,NCLPDB                                                  000531
      DBPOLE(I,1)=CLPOLE(I,1)                                            000532
  150 DBPOLE(I,2)=CLPOLE(I,2)                                            000533
      IF(NCLZDB .EQ. 0) GO TO 152                                        000534
      DO 151 I=1,NCLZDB                                                  000535
      DBZERO(I,1)=CLZERO(I,1)                                            000536
  151 DBZERO(I,2)=CLZERO(I,2)                                            000537
  152 CONTINUE                                                           000538
C                                                                        000539
C                                                                        000540
C   WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW     000541
      IF(TXFORM .EQ.2HUP .OR. TXFORM .EQ.2HW ) GO TO 5000                000542
C   WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW     000543
C                                                                        000544
C                                                                        000545
C   ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ     000546
C   ZZZZZZZZZZZZ  ADD POLE / CANCEL ZERO AT Z=0.0  ZZZZZZZZZZZZZZ       000547
C   ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ     000548
C                                                                        000549
C   EITHER CANCEL ZERO AT Z=0.0 IN NUMERATOR POLY (DBNPOLY)              000550
```

```
C    OR ADD POLE AT Z=0.0 IN DENOMINATOR POLY (DBDPOLY).                    000551
C    SET IADPOLE = 10 IF POLE ADDED AT Z=0.0                                000552
C                                                                           000553
C   CHECK LAST Z-POWER IN NUMBERATOR FOR 0                                  000554
      IF(DBNPOLY(NN)) 100,115,100                                           000555
C  REDUCE NUMERATOR Z-PWRS BY 1 TO CANCEL Z=0.0                             000556
  100 CONTINUE                                                              000557
      DO 122 I=2,NN,2                                                       000558
  122 DBNPOLY(I)=DBNPOLY(I)-1                                               000559
      GO TO 134                                                             000560
C  INCREASE DENOMINATOR Z-PWRS BY 1 TO ADD Z=0                              000561
  115 CONTINUE                                                              000562
C   SET IADPOLE = 10 TO FLAG ADDED POLE AT Z=0.0                            000563
      IADPOLE=10                                                            000564
      DO 147 I=2,ND,2                                                       000565
  147 DBDPOLY(I)=DBDPOLY(I)+1                                               000566
  134 CONTINUE                                                              000567
C                                                                           000568
      NCLPDB=NCLP                                                           000569
C   SKIP THIS SECTION IF IADPOLE = 0                                        000570
      IF(IADPOLE-1) 165,165,173                                             000571
  173 NCLPDB=NCLP+1                                                         000572
      DBPOLE(NCLPDB,1)=0.0                                                  000573
      DBPOLE(NCLPDB,2)=0.0                                                  000574
  165 CONTINUE                                                              000575
C                                                                           000576
C                                                                           000577
C    ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ   000578
      GO TO 5100                                                            000579
C    ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ   000580
C                                                                           000581
C                                                                           000582
C                                                                           000583
 5000 CONTINUE                                                              000584
C                                                                           000585
C     UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU   000586
C     UUUUUUUUUUUUUUUU  (AP+WP)  FACTOR SECTION  UUUUUUUUUUUUUUUUUUUUUU   000587
C     UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU   000588
C                                                                           000589
C     THIS SECTION HUNTS FOR A HIGH-RATE NUMERATOR ROOT                     000590
C     (AP + WP) IN N(WP) AND EFFECTIVELY CANCELS IT IN                      000591
C     THE NUMERATOR INSTEAD OF INCLUDING IT AS A                            000592
C     DENOMINATOR POLE AND A FACTOR IN THE DENOMINATOR                      000593
C     POLYNOMIAL D(WP).  IN REALITY, THE (AP +WP) ROOT IS                   000594
C     NOT CANCELLED IN THE NUMERATOR BUT IS CARRIED ALONG                   000595
C     AS AN EQUAL NUMERATOR AND DENOMINATOR FACTOR THAT                     000596
C     CANCELS DURING THE RESIDUE CALCULATIONS.  THIS                        000597
C     METHOD OF HANDLING THE NUMERATOR ZERO REMOVES THE                     000598
C     REQUIREMENT OF ACTUALLY CANCELLING THE ZERO AND                       000599
C     MULTIPLYING THE REMAINING ZEROS TO FORM THE NEW                       000600
C     NUMERATOR POLYNOMIAL.  SUCH A PROCEDURE OF                            000601
C     MULTIPLYING ROOTS,  WHICH MAY BE IN ERROR TO BEGIN                    000602
C     WITH, TENDS TO ADD NUMERICAL ERRORS TO THE                            000603
C     OVERALL CALCULATIONS.                                                 000604
C                                                                           000605
C     THE VARIABLE 'KZERO' IS SET EQUAL TO 10 TO INDICATE
```

```
C     THAT THE (AP + UP) ROOT APPEARS IN THE NUMERATOR                  000606
C     AND SHOULD BE INCLUDED AS AN CANCELLING FACTOR IN                 000607
C     THE RESIDUE CALCULATIONS.                                         000608
C                                                                       000609
C     THE (AP +UP) FACTOR IS MULTIPLIED BY THE ORIGINALLY               000610
C     INPUTTED DENOMINATOR POLYNOMIAL D(UP) AND ADDED AS                000611
C     A DENOMINATOR POLE IF THIS FACTOR IS NOT A ROOT IN                000612
C     THE NUMERATOR.  THE RESIDUE OF THE (AP + UP) POLE                 000613
C     IS THEN CALCULATED ALONG WITH THE OTHER POLES OF D(UP).           000614
C                                                                       000615
C     THE OVERALL RESIDUE EXPRESSION FOR THE U OR U' PLANE              000616
C     IS GIVEN BY:                                                      000617
C                                                                       000618
C     NUMERATOR = 2AP*[N(UP)/D*(P)]*[(A+U)/(1+Y**XNT)]                  000619
C     DENOMINATOR = U +  A*[(1-Y**XNT)/(1+Y**XNT)]                      000620
C                                                                       000621
C     WHERE                                                             000622
C                                                                       000623
C          D*(P) = D(UP) * (AP+UP) * (AP-UP)                            000624
C          Y = [(AP+UP)/(AP-UP)]                                        000625
C          Y**XNT = [(AP+UP)/(AP-UP)]**XNT                              000626
C                                                                       000627
C          U = LOW-RATE  U OR U' VARIABLE [Z = EXP(ST)]                 000628
C          UP = HIGH-RATE  U OR U' VARIABLE [Z = EXP(ST/N)]             000629
C                                                                       000630
C          BILINEAR TRANSFORMATION BETWEEN Z, U, AND U' PLANES:         000631
C              U = (Z-1)/(Z+1)              U' = (2/T)*(Z-1)/(Z+1)      000632
C              Z = (1+U)/(1-U)              Z = (2/T+U )/(2/T-U')       000633
C                                                                       000634
C                                                                       000635
C                                                                       000636
C          U TRANSFORM:                                                 000637
C                      A=1.0                                            000638
C                      AP=1.0                                           000639
C                      U = U                                            000640
C                      UP = UP                                          000641
C                                                                       000642
C          U' TRANSFORM:                                                000643
C                       A = 2/TEXT                                      000644
C                       AP = 2/TINT                                     000645
C                       U = U'                                          000646
C                       UP = UP'                                        000647
C                                                                       000648
C          TEXT = LOW-RATE SAMPLING PERIOD (SEC)                        000649
C          TINT = HIGH-RATE SAMPLING PERIOD (SEC)                       000650
C          XNT = TEXT/TINT                                              000651
C                                                                       000652
C                                                                       000653
      KZERO=0                                                           000654
C  SET XAP VARIABLE TO U OR U' ZERO AT  UP = -AP                        000655
      XAP= -1.0                                                         000656
      DXAP= -1.0                                                        000657
      IF(TXFORM .EQ. 2HUP) XAP= -2.0/TINT                               000658
      IF(TXFORM .EQ. 2HUP) DXAP= -2.0/TINT                              000659
C  DO LOOP TO HUNT FOR (AP+UP) ZERO IN NUMERATOR                        000660
```

```
      IF(NCLZDB .EQ. 0) GO TO 4001                                      000661
      DO 4000 I=1,NCLZDB                                                000662
      A1=DBZERO(I,1)                                                    000663
      A2=DBZERO(I,2)                                                    000664
C  CHECK FOR ZERO AT (AP+UP).  IF FOUND, SET THE KZERO                  000665
C  VARIABLE TO "10" TO INVOKE THE CANCELLATION OF THIS                  000666
C  ZERO DURING THE RESIDUE CALCULATIONS.                                000667
      IF(ABS(A1-XAP).LE.1.E-11.AND.ABS(A2).LE.1.E-11) GO TO 5010        000668
 4000 CONTINUE                                                          000669
 4001 CONTINUE                                                          000670
C  NO ZERO AT (AP+UP) WAS FOUND.  ADD DOUBLE PRECISION POLE AT          000671
C                       (UP = -AP)                                      000672
      NCLPDB=NCLPDB+1                                                   000673
      DBPOLE(NCLPDB,1)= DXAP                                            000674
      DBPOLE(NCLPDB,2)= 0.0                                             000675
C  MULTIPLY THE G(UP) DENOMINATOR TIMES (AP+UP).                        000676
C  THAT IS,  D(UP) * (AP+UP)                                            000677
      AP(1) = 1.0                                                       000678
      AP(2) = 1.0                                                       000679
      AP(3) = -DXAP                                                     000680
      AP(4) = 0.0                                                       000681
      CALL MULTIP(AP,4,DBDPOLY,ND,A,NT3,2)                              000682
C  TRANSFER NEW DENOMINATOR POLY TO DBDPOLY                             000683
      DO 5020 I=1,NT3                                                   000684
 5020 DBDPOLY(I)=A(I)                                                   000685
C  SET DBDPOLY ELEMENTS, I.E., DBDPOLY(ND)                              000686
      ND=NT3                                                            000687
C  ORDER DBDPOLY IN DESCENDING POWERS OF U OR U'                        000688
      CALL ORDER3(DBDPOLY,ND,2)                                         000689
      GO TO 5030                                                        000690
C                                                                       000691
C  SET KZERO VARIABLE TO INDICATE (AP+UP) ZERO PRESENT                  000692
 5010 KZERO = 10                                                        000693
 5030 CONTINUE                                                          000694
C                                                                       000695
C                                                                       000696
C     UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU  000697
C     UUUUUUUUUUUUUUUU  (AP-UP)  FACTOR SECTION  UUUUUUUUUUUUUUUUU     000698
C     UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU  000699
C                                                                       000700
C     THIS SECTION MULTIPLIES THE DENOMINATOR POLYNOMIAL                000701
C     D(UP) BY THE FACTOR (AP-UP) PRIOR TO TAKING THE                   000702
C     DERIVATIVE WITH RESPECT TO THE HIGH-RATE U OR U'                  000703
C     TRANSFORM VARIABLE UP.  THE INCLUSION OF THE                      000704
C     (AP-UP) FACTOR OR POLE IN THE DENOMINATOR POLYNOMIAL              000705
C     IS NOT ACTUALLY NECESSARY  SINCE THE (AP-UP) POLE                 000706
C     IS NOT INCLUDED IN THE INTRGRATION CONTOUR THAT IS                000707
C     BEING USED.  INCLUDING THE (AP-UP) FACTOR IN THE                  000708
C     DENOMINATOR D(UP) OR CARRING IT ALONG AS A SEPARATE               000709
C     FACTOR HAS THE SAME EFFECT ON THE RESIDUES.                       000710
C                                                                       000711
      AP(1) = -1.0                                                      000712
      AP(2) = 1.0                                                       000713
      AP(3) = -DXAP                                                     000714
      AP(4) = 0.0                                                       000715
```

```
      CALL MULTIP(AP,4,DBDPOLY,ND,A,NT3,2)                              000716
C  TRANSFER NEW DENOMINATOR POLYNOMIAL TO DBDPOLY                        000717
      DO 5040 I=1,NT3                                                   000718
 5040 DBDPOLY(I)=A(I)                                                   000719
C  SET DBDPOLY ELEMENTS .I.E.,  DBDPOLY(ND)                            000720
      ND=NT3                                                            000721
C  ORDER DBDPOLY IN DESCENDING POWERS OF W OR W'                        000722
      CALL ORDER3(DBDPOLY,ND,2)                                         000723
C                                                                       000724
C                                                                       000725
 5100 CONTINUE                                                          000726
C                                                                       000727
C                                                                       000728
C  OBTAIN MULTIPLICITY OF EACH POLE AND DELETE EXTRA POLES              000729
C                                                                       000730
      IROW=NORDER+1                                                     000731
      ICOL=2                                                            000732
      CALL ORDPOLE(DBPOLE,IROW,ICOL,NCLPDB,NPOLES,MULPOLE,NM)           000733
C                                                                       000734
CC                                                                      000735
C     ***************************************************************   000736
C     ****************  WRITE SECTION FOR INPUT *********************   000737
C     ***************************************************************   000738
C                                                                       000739
      WRITE(7,7) TXFORM                                                 000740
    7 FORMAT(1H ,///,14X,A2,*-PLANE  HIGH-RATE POLES*,/)                000741
      WRITE(7,15) (DBPOLE(I,1),DBPOLE(I,2),I=1,NM)                      000742
   15 FORMAT(1H ,5X,*(*,G18.10,*,*,G18.10,* )*)                         000743
      WRITE(7,10)                                                       000744
   10 FORMAT(1H ,* *)                                                   000745
      WRITE(7,17) (MULPOLE(I),I=1,NM)                                   000746
   17 FORMAT(1H ,16X,* MULTIPLICITY =*,I3,)                             000747
C                                                                       000748
C                                                                       000749
C    ***********************************************************        000750
C    **********  MASTER DO LOOP FOR RESIDUES  **************           000751
C    ***********************************************************        000752
C                                                                       000753
C    THE LOW-RATE Z-PLANE TRANSFORM IS OBTAINED FROM THE                000754
C    HIGH-RATE Z-PLANE TRANSFORM USING THE FOLLOWING                    000755
C    GENERAL EXPRESSION:                                                000756
C                                                                       000757
C      G(Z) = SUM RESIDUES [ G(P)/P * (Z)/(Z-P**XNT) ]                  000758
C             [ P=POLES G(P)/P ]                                        000759
C                                                                       000760
C      G(Z) = SUM RESIDUES [ [N(P)*Z] / [D1(P)*Z - D1(P)*P**XNT] ]      000761
C             [ P=POLES G(P)/P ]                                        000762
C                                                                       000763
C          G(Z) = N(Z)/D(Z)  ---  LOW-RATE TXFORM  Z=EXP(S*T)           000764
C          G(P) = N(P)/D(P)  ---  HIGH-RATE TXFORM  P=Z=EXP(S*T/N)      000765
C          D1(P) = D(P) * P                                             000766
C          T = LOW-RATE SAMPLING INTERVAL                               000767
C          T/N = HIGH-RATE SAMPLING INTERVAL                            000768
C          XNT = (T) / (T/N) = N                                        000769
C                                                                       000770
```

```
C   EACH Z-PLANE DESIDUE IS CALCULATED BY SEPARATE SUBROUTINES          000771
C   ACCORDING TO THE MULTIPLICITY.                                      000772
C         ZMULT1    ----   MULTIPLICITY = 1                           000773
C         ZMULT2    ----   MULTIPLICITY = 2                           000774
C         ZMULT3    ----   MULTIPLICITY = 3                           000775
C                                                                       000776
C   IN THE RESIDUE EXPRESSIONS MECHANIZED IN THESE SUBROUTINES,         000777
C   THE REQUIRED DERIVATIVES WITH RESPECT TO THE INTEGRATION            000778
C    VARIABLE  'P' OR 'WP'  ARE OBTAINED IN A NON-CONVENTIONAL          000779
C   MANNER.  SEPARATE NUMERATOR AND DENOMINATOR DERIVATIVES ARE         000780
C   CALCULATED WITHOUT FIRST CANCELLING THE POLE UNDER                  000781
C   EVALUATION IN THE DENOMINATOR.  THIS RESULTS IN AN                  000782
C   INDETERMINATE FORM WHICH MUST BE EVALUATED USING LA'HOSPITALS       000783
C   RULE.  THE PROCEDURE THAT IS FOLLOWED IS TO TAKE CONSEQUETIVE       000784
C   DERIVATIVES OF THE NUMERATOR AND DENOMINATOR UNTILL GENERAL         000785
C   RATIONAL EXPRESSIONS ARE OBTAINED FOR THE RESIDUES.                 000786
C                                                                       000787
C                                                                       000788
C   THE W OR W' PLANE LOW-RATE TRANSFORM IS OBTAINED FROM THE           000789
C   W OR W' PLANE HIGH-RATE TRANSFORM USING THE EXPRESSION              000790
C   GIVEN IN THE "(AP+WP) FACTOR SECTION" AND REPEATED IN A MORE        000791
C   GENERAL FORM BELOW.                                                 000792
C                                                                       000793
C   G(W) = SUM RESIDUES [AP*[G(WP)/(AP+WP)(AP-WP)]*[1/1-X]              000794
C                              WP = POLES OF G(WP)/(AP+WP)              000795
C                                                                       000796
C   WHERE                                                               000797
C                                                                       000798
C       X = [(AP+WP)/(AP-WP)]**XNT    *    [(A+W)/(A-W)]**-1          000799
C       G(WP) = N(WP)/D(WP)    -----  HIGH-RATE TRANSFER FUNCTION       000800
C       G(W) = N(W)/D(W)    -----  LOW-RATE TRANSFER FUNCTION           000801
C                                                                       000802
C       W TRANSFORM:                                                    000803
C                  A=1.0                                                000804
C                  AP=1.0                                               000805
C                  W=W                                                  000806
C                  WP=WP                                                000807
C       W' TRANSFORM:                                                   000808
C                  A=2/TEXT                                             000809
C                  AP=2/TINT                                            000810
C                  W=W'                                                 000811
C                  WP=WP'                                               000812
C                                                                       000813
C       TEXT = LOW-RATE SAMPLING PERIOD (SEC)                           000814
C       TINT = HIGH-RATE SAMPLING PERIOD (SEC)                          000815
C       XNT = TEXT/TINT                                                 000816
C                                                                       000817
C   EACH W-PLANE OR W'-PLANE RESIDUE IS CALCULATED BY SEPARATE          000818
C   SUBROUTINES ACCORDING TO THE MULTIPLICITY.                          000819
C       WMULT1    ----    MULTIPLICITY = 1                              000820
C       WMULT2    ----    MULTIPLICITY = 2                              000821
C       WMULT3    ----    MULTIPLICITY = 3                              000822
C                                                                       000823
C                                                                       000824
C                                                                       000825
```

```
C  INITIALIZE RESIDUE COUNTERS TO ZERO                                    000826
      JRES1=0                                                             000827
      JRES2=0                                                             000828
      JRES3=0                                                             000829
C  START OF MASTER DO LOOP FOR RESIDUES                                   000830
C  NM = NUMBER OF DISTINCT POLES                                          000831
      DO 400 I=1,NM                                                       000832
C  SET COUNTER VARIABLE FOR RESIDUE SUBROUTINES                           000833
      IPOLE=I                                                             000834
C  CHECK FOR MULTIPLICITY .GT. 3                                          000835
      IF(MULPOLE(IPOLE) .GT. 3) GO TO 999                                 000836
C  BRANCH ACCORDING TO MULTIPLICITY OF POLE                               000837
      GO TO (1000,2000,3000), MULPOLE(IPOLE)                              000838
  999 WRITE(7,1)                                                          000839
    1 FORMAT(1H ,*MULTIPLICITY GREATER THAN 3  - EXIT PROGRAM *)          000840
      STOP                                                                000841
 1000 IF(TXFORM .EQ. 2HUP .OR. TXFORM .EQ. 2HU ) GO TO 1001               000842
C                                                                         000843
C                                                                         000844
C-------------------------------------------------------------------------000845
      CALL OVERLAY(6HTXCONV,23,1)                                         000846
C-------------------------------------------------------------------------000847
C                                                                         000848
C-------------------------------------------------------------------------000849
C     CALL ZMULT1                                                         000850
C-------------------------------------------------------------------------000851
C                                                                         000852
      GO TO 400                                                           000853
C                                                                         000854
C-------------------------------------------------------------------------000855
 1001 CALL OVERLAY(6HTXCONV,23,2)                                         000856
C-------------------------------------------------------------------------000857
C                                                                         000858
C-------------------------------------------------------------------------000859
C1001 CALL WMULT1                                                         000860
C-------------------------------------------------------------------------000861
C                                                                         000862
      GO TO 400                                                           000863
 2000 IF(TXFORM .EQ. 2HUP .OR. TXFORM .EQ. 2HU ) GO TO 2001               000864
      WRITE(7,2010)                                                       000865
 2010 FORMAT(1H ,*MULTIPLICITY .EQ. 2 IN ZT-PLANE - EXIT PROGRAM*)        000866
      STOP                                                                000867
C                                                                         000868
C-------------------------------------------------------------------------000869
 2001 CALL OVERLAY(6HTXCONV,23,3)                                         000870
C-------------------------------------------------------------------------000871
C                                                                         000872
C-------------------------------------------------------------------------000873
C2001 CALL WMULT2                                                         000874
C-------------------------------------------------------------------------000875
C                                                                         000876
      GO TO 400                                                           000877
 3000 IF(TXFORM .EQ. 2HUP .OR. TXFORM .EQ. 2HU ) GO TO 3001               000878
      WRITE(7,2011)                                                       000879
 2011 FORMAT(1H ,*MULTIPLICITY .EQ. 3 IN ZT-PLANE - EXIT PROGRAM*)        000880
```

```
      STOP                                                                  000881
C                                                                           000882
C-----------------------------------------------------------------------    000883
 3001 CALL OVERLAY(6HTXCONV,23,4)                                           000884
C-----------------------------------------------------------------------    000885
C                                                                           000886
C-----------------------------------------------------------------------    000887
C3001 CALL WMULT3                                                           000888
C-----------------------------------------------------------------------    000889
C                                                                           000890
  400 CONTINUE                                                              000891
C                                                                           000892
C                                                                           000893
      IF(JRES1 .EQ. 0) GO TO 401                                            000894
C                                                                           000895
C-----------------------------------------------------------------------    000896
      CALL OVERLAY(6HTXCONV,23,5)                                           000897
C-----------------------------------------------------------------------    000898
C                                                                           000899
C-----------------------------------------------------------------------    000900
C     CALL RES1                                                             000901
C-----------------------------------------------------------------------    000902
C                                                                           000903
C                                                                           000904
      IF(KTXFORM .LT. 5) GO TO 401                                          000905
      IF(JRES1 .EQ. 0) GO TO 401                                            000906
C                                                                           000907
C  CALL WZBILIN TO TRANSFER W' POLY AND ROOTS BACK INTO Z-PLANE              000908
      NRD=NORDER+1                                                          000909
      NCD=2                                                                 000910
      ALPHA=2.0/TEXT                                                        000911
      SIGMA1=-1.0                                                           000912
      SIGMA2=1.0                                                            000913
      KEY=4                                                                 000914
      T=TEXT                                                                000915
      CALL WZBILIN(RES1N,KRES1N,RES1D,KRES1D,                               000916
     1 RES1POL,JRES1,NRD,NCD,ALPHA,SIGMA1,SIGMA2,3,T,KEY,ZFORM)             000917
  401 CONTINUE                                                              000918
C                                                                           000919
C                                                                           000920
      IF(JRES2 .EQ. 0) GO TO 402                                            000921
C                                                                           000922
C-----------------------------------------------------------------------    000923
      CALL OVERLAY(6HTXCONV,23,6)                                           000924
C-----------------------------------------------------------------------    000925
C                                                                           000926
C-----------------------------------------------------------------------    000927
C     CALL RES2                                                             000928
C-----------------------------------------------------------------------    000929
C                                                                           000930
C                                                                           000931
      DO 412 I=1,JRES2                                                      000932
      DBPOLE(I,1)=RES2POL(I,1)                                              000933
  412 DBPOLE(I,2)=RES2POL(I,2)                                              000934
      IF(KTXFORM .LT. 5) GO TO 402                                          000935
```

```
C                                                                       000936
C  CALL I    .IN TO TRANSFER W' POLY AND ROOTS BACK INTO Z-PLANE         000937
      NR    RDER+1                                                      000938
      N'                                                                000939
      A=2.0/TEXT                                                        000940
      .MA1=-1.0                                                         000941
      SIGMA2=1.0                                                        000942
      KEY=4                                                             000943
      T=TEXT                                                            000944
C                                                                       000945
      CALL WZBILIN(RES2N,KRES2N,RES2D,KRES2D,                           000946
     1 DBPOLE,JRES2,NRD,NCD,ALPHA,SIGMA1,SIGMA2,3,T,KEY,ZFORM)          000947
      IF(ZFORM .EQ. 2HZR) GO TO 402                                     000948
      KJRES2=10                                                         000949
  402 CONTINUE                                                          000950
C                                                                       000951
C                                                                       000952
      IF(JRES3 .EQ. 0) GO TO 403                                        000953
C                                                                       000954
C-----------------------------------------------------------------------000955
      CALL OVERLAY(6HTXCONV,23,7)                                       000956
C-----------------------------------------------------------------------000957
C                                                                       000958
C-----------------------------------------------------------------------000959
C     CALL RES3                                                         000960
C-----------------------------------------------------------------------000961
C                                                                       000962
C                                                                       000963
      DO 413 I=1,JRES3                                                  000964
      DBZERO(I,1)=RES3POL(I,1)                                          000965
  413 DBZERO(I,2)=RES3POL(I,2)                                          000966
      IF(KTXFORM .LT. 5) GO TO 403                                      000967
C                                                                       000968
C  CALL WZBILIN TO TRANSFER W' POLYS AND ROOTS BACK INTO Z-PLANE        000969
      NRD=NORDER+1                                                      000970
      NCD=2                                                             000971
      ALPHA=2.0/TEXT                                                    000972
      SIGMA1=-1.0                                                       000973
      SIGMA2=1.0                                                        000974
      KEY=4                                                             000975
      T=TEXT                                                            000976
      CALL WZBILIN(RES3N,KRES3N,RES3D,KRES3D,                           000977
     1 DBZERO,JRES3,NRD,NCD,ALPHA,SIGMA1,SIGMA2,3,T,KEY,ZFORM)          000978
      IF(ZFORM .EQ. 2HZR) GO TO 403                                     000979
      KJRES3=10                                                         000980
  403 CONTINUE                                                          000981
C                                                                       000982
C                                                                       000983
C     ************************************************************      000984
C     **********  LOW-RATE  TRANSFER FUNCTION ADDITION  **********      000985
C     ************************************************************      000986
C                                                                       000987
C     ADD THE INDIVIDUAL LOW-RATE TRANSFER FUNCTIONS FOR THE 1ST,       000988
C     2ND AND 3RD ORDER RESIDUES TO OBTAIN THE OUTPUT LOW-RATE          000989
C     TRANSFER FUNCTION.                                                000990
```

```
C                                                                       000991
C  FORM THE COMBINED NUMERATOR POLYNOMIALS                              000992
C                                                                       000993
C           RES1N=RES1N*RES2D*RES3D                                     000994
C           RES2N=RES2N*RES1D*RES3D                                     000995
C           RES3N=RES3N*RES1D*RES2D                                     000996
C                                                                       000997
      IF(KRES1N .EQ. 0) GO TO 800                                       000998
      IF(JRES2 .EQ. 0)  GO TO 805                                       000999
      CALL MULTIP(RES1N,KRES1N,RES2D,KRES2D,C3,NT3,3)                   001000
      CALL DOLOOP(C3,NT3,RES1N,KRES1N)                                  001001
  805 IF(JRES3 .EQ. 0) GO TO 800                                        001002
      CALL MULTIP(RES1N,KRES1N,RES3D,KRES3D,C3,NT3,3)                   001003
      CALL DOLOOP(C3,NT3,RES1N,KRES1N)                                  001004
  800 CONTINUE                                                          001005
      IF(JRES2 .EQ. 0) GO TO 810                                        001006
      IF(KRES1N .EQ. 0) GO TO 815                                       001007
      CALL MULTIP(RES2N,KRES2N,RES1D,KRES1D,C3,NT3,3)                   001008
      CALL DOLOOP(C3,NT3,RES2N,KRES2N)                                  001009
  815 IF(JRES3 .EQ. 0) GO TO 810                                        001010
      CALL MULTIP(RES2N,KRES2N,RES3D,KRES3D,C3,NT3,3)                   001011
      CALL DOLOOP(C3,NT3,RES2N,KRES2N)                                  001012
  810 CONTINUE                                                          001013
C                                                                       001014
      IF(JRES3 .EQ. 0) GO TO 820                                        001015
      IF(KRES1N .EQ. 0) GO TO 825                                       001016
      CALL MULTIP(RES3N,KRES3N,RES1D,KRES1D,C3,NT3,3)                   001017
      CALL DOLOOP(C3,NT3,RES3N,KRES3N)                                  001018
  825 IF(JRES2 .EQ. 0) GO TO 820                                        001019
      CALL MULTIP(RES3N,KRES3N,RES2D,KRES2D,C3,NT3,3)                   001020
      CALL DOLOOP(C3,NT3,RES3N,KRES3N)                                  001021
  820 CONTINUE                                                          001022
C                                                                       001023
C                                                                       001024
C  ADD THE COMBINED NUMERATOR POLYS [RES1N=RES1N+RES2N+RES3N]           001025
C  AND FORM THE COMMON DENOMINATOR POLY [RES1D=RES1D*RES2D*RES3D]       001026
C                                                                       001027
      IF(KRES1N .EQ. 0) GO TO 830                                       001028
      IF(JRES2 .EQ. 0)  GO TO 840                                       001029
      IF(JRES3 .EQ. 0)  GO TO 850                                       001030
C                                                                       001031
      CALL ADD(RES1N,KRES1N,RES2N,KRES2N,C3,NT3,3)                      001032
      CALL ADD(C3,NT3,RES3N,KRES3N,C6,NT6,3)                            001033
      CALL DOLOOP(C6,NT6,RES1N,KRES1N)                                  001034
      CALL MULTIP(RES1D,KRES1D,RES2D,KRES2D,C3,NT3,3)                   001035
      CALL MULTIP(C3,NT3,RES3D,KRES3D,C6,NT6,3)                         001036
      CALL DOLOOP(C6,NT6,RES1D,KRES1D)                                  001037
      GO TO 855                                                         001038
C                                                                       001039
  830 IF(JRES2 .EQ. 0) GO TO 832                                        001040
      IF(JRES3 .EQ. 0) GO TO 834                                        001041
      CALL ADD(RES2N,KRES2N,RES3N,KRES3N,C3,NT3,3)                      001042
      CALL DOLOOP(C3,NT3,RES1N,KRES1N)                                  001043
      CALL MULTIP(RES2D,KRES2D,RES3D,KRES3D,C3,NT3,3)                   001044
      CALL DOLOOP(C3,NT3,RES1D,KRES1D)                                  001045
```

```
      GO TO 855                                                          001046
  832 CALL DOLOOP(RES3N,KRES3N,RES1N,KRES1N)                             001047
      CALL DOLOOP(RES3D,KRES3D,RES1D,KRES1D)                             001048
      GO TO 855                                                          001049
  834 CALL DOLOOP(RES2N,KRES2N,RES1N,KRES1N)                             001050
      CALL DOLOOP(RES2D,KRES2D,RES1D,KRES1D)                             001051
      GO TO 855                                                          001052
C                                                                        001053
  840 IF(JRES3 .EQ. 0) GO TO 855                                         001054
      CALL ADD(RES1N,KRES1N,RES3N,KRES3N,C3,NT3,3)                       001055
      CALL DOLOOP(C3,NT3,RES1N,KRES1N)                                   001056
      CALL MULTIP(RES1D,KRES1D,RES3D,KRES3D,C3,NT3,3)                    001057
      CALL DOLOOP(C3,NT3,RES1D,KRES1D)                                   001058
      GO TO 855                                                          001059
C                                                                        001060
  850 CALL ADD(RES1N,KRES1N,RES2N,KRES2N,C3,NT3,3)                       001061
      CALL DOLOOP(C3,NT3,RES1N,KRES1N)                                   001062
      CALL MULTIP(RES1D,KRES1D,RES2D,KRES2D,C3,NT3,3)                    001063
      CALL DOLOOP(C3,NT3,RES1D,KRES1D)                                   001064
C                                                                        001065
  855 CONTINUE                                                           001066
C                                                                        001067
C                                                                        001068
C  ELIMINATE ZERO LEADING COEFFICIENT IN NUMERATOR POLY                  001069
C                                                                        001070
      II=1                                                               001071
  607 DO 609 I=II,KRES1N,3                                               001072
      IF(DABS(RES1N(I)) .GE. 1.D-11) GO TO 610                           001073
      KRES1N=KRES1N-3                                                    001074
      IF(KRES1N .EQ. 0) GO TO 610                                        001075
      DO 608 J=I,KRES1N                                                  001076
  608 RES1N(J)=RES1N(J+3)                                                001077
      II=I                                                               001078
      GO TO 607                                                          001079
  609 CONTINUE                                                           001080
  610 CONTINUE                                                           001081
C                                                                        001082
C  ELIMINATE ZERO LEADING COEFFCIENTS IN DENOMINATOR POLY                001083
C                                                                        001084
      II=1                                                               001085
  611 DO 613 I=II,KRES1D,3                                               001086
      IF(RES1D(I) .NE. 0.0) GO TO 614                                    001087
      KRES1D=KRES1D-3                                                    001088
      IF(KRES1D .EQ. 0) GO TO 614                                        001089
      DO 612 J=I,KRES1D                                                  001090
  612 RES1D(J)=RES1D(J+3)                                                001091
      II=I                                                               001092
      GO TO 611                                                          001093
  613 CONTINUE                                                           001094
  614 CONTINUE                                                           001095
C                                                                        001096
C                                                                        001097
C  ORDER THE TOTAL NUMERATOR AND DENOMINATOR POLYNOMIALS IN              001098
C  DESCENDING POWERS AND ADD MISSING COEFFICIENTS                        001099
C                                                                        001100
```

```
      IF(KRES1N .EQ. 3) GO TO 605                                       001101
      CALL ORDER3(RES1N,KRES1N,3)                                       001102
  605 IF(KRES1D .EQ. 3) GO TO 606                                       001103
      CALL ORDER3(RES1D,KRES1D,3)                                       001104
  606 NCLZ=RES1N(3)                                                     001105
      NCLP=RES1D(3)                                                     001106
C     NCLZ1=NCLZ                                                        001107
C     NCLP1=NCLP                                                        001108
      IF(NCLZ .EQ. 0) GO TO 604                                         001109
      CALL COEFF(RES1N,KRES1N,NCLZ,3)                                   001110
  604 CALL COEFF(RES1D,KRES1D,NCLP,3)                                   001111
C                                                                       001112
C  NORMALIZE NUMERATOR AND DENOMINATOR POLYS AND TRANSFER               001113
C  REAL COEFFICIENTS (ONLY) INTO ARRAYS CLNPOL1 AND CLDPOL1.            001114
C                                                                       001115
      K=1                                                               001116
      CLNK=RES1N(1)                                                     001117
      CLDK=RES1D(1)                                                     001118
      DBCLK=DBCLK*(CLNK/CLDK)                                           001119
      DO 870 I=1,KRES1N,3                                               001120
      IF(DABS(RES1N(I)/CLNK) .LE. 1.D-20) RES1N(I)=0.0                  001121
      IF(DABS(RES1N(I+1)/CLNK) .LE. 1.D-20) RES1N(I+1)=0.0              001122
C     CLNPOL1(K)=RES1N(I)/CLDK                                          001123
      RES1N(I)=RES1N(I)/CLNK                                            001124
      RES1N(I+1)=RES1N(I+1)/CLNK                                        001125
  870 K=K+1                                                             001126
C                                                                       001127
      K=1                                                               001128
      DO 871 I=1,KRES1D,3                                               001129
      IF(DABS(RES1D(I)/CLDK) .LE. 1.D-20) RES1D(I)=0.0                  001130
      IF(DABS(RES1D(I+1)/CLDK) .LE. 1.D-20) RES1D(I+1)=0.0              001131
C     CLDPOL1(K)=RES1D(I)/CLDK                                          001132
      RES1D(I)=RES1D(I)/CLDK                                            001133
      RES1D(I+1)=RES1D(I+1)/CLDK                                        001134
  871 K=K+1                                                             001135
C                                                                       001136
C     CLNK1=CLNPOL1(1)                                                  001137
C     CLK1=CLNPOL1(1)                                                   001138
C     CLDK1=CLDPOL1(1)                                                  001139
C                                                                       001140
C  OBTAIN THE LOW-RATE NUMERATOR ROOTS (ZEROS)                          001141
C                                                                       001142
      IF(NCLZ .EQ. 0) GO TO 876                                         001143
      K=1                                                               001144
      DO 874 I=1,KRES1N,3                                               001145
      A(K)=RES1N(I)                                                     001146
      B(K)=RES1N(I+1)                                                   001147
  874 K=K+1                                                             001148
      CALL ROOTS(A,B,NCLZ,C3,C6)                                        001149
      DO 875 I=1,NCLZ                                                   001150
      IF(DABS(C3(I)) .LE. 1.D-20) C3(I)=0.0                             001151
      IF(DABS(C6(I)) .LE. 1.D-20) C6(I)=0.0                             001152
      CLZERO(I,1)=C3(I)                                                 001153
      CLZERO(I,2)=C6(I)                                                 001154
C     CLZERO1(I,1)=C3(I)                                                001155
```

```
C      CLZERO1(I,2)=C6(I)                                                  001156
  875 CONTINUE                                                             001157
  876 CONTINUE                                                             001158
C                                                                          001159
C                                                                          001160
C  TRANSFER LOW-RATE POLES INTO ARRAY CLPOLE                               001161
C                                                                          001162
       IF(JRES1 .EQ. 0) GO TO 881                                          001163
       DO 880 I=1,JRES1                                                    001164
       IF(DABS(RES1POL(I,1)) .LE. 1.D-20) RES1POL(I,1)=0.0                 001165
       IF(DABS(RES1POL(I,2)) .LE. 1.D-20) RES1POL(I,2)=0.0                 001166
       CLPOLE(I,1)=RES1POL(I,1)                                            001167
       CLPOLE(I,2)=RES1POL(I,2)                                            001168
C      CLPOLE1(I,1)=RES1POL(I,1)                                           001169
C      CLPOLE1(I,2)=RES1POL(I,2)                                           001170
  880 CONTINUE                                                             001171
C                                                                          001172
  881 IF(JRES2 .EQ. 0) GO TO 887                                           001173
       KK=JRES1+1                                                          001174
       KP=2                                                                001175
       IF(KJRES2 .GT. 5) KP=1                                              001176
       DO 886 K=1,JRES2                                                    001177
       IF(DABS(DBPOLE(K,1)) .LE. 1.D-20) DBPOLE(K,1)=0.0                   001178
       IF(DABS(DBPOLE(K,2)) .LE. 1.D-20) DBPOLE(K,2)=0.0                   001179
       DO 885 I=1,KP                                                       001180
       CLPOLE(KK,1)=DBPOLE(K,1)                                            001181
       CLPOLE(KK,2)=DBPOLE(K,2)                                            001182
C      CLPOLE1(KK,1)=DBPOLE(K,1)                                           001183
C      CLPOLE1(KK,2)=DBPOLE(K,2)                                           001184
  885 KK=KK+1                                                              001185
  886 CONTINUE                                                             001186
C                                                                          001187
  887 IF(JRES3 .EQ. 0) GO TO 892                                           001188
       KK=JRES1+(KP*JRES2)+1                                               001189
       KP=3                                                                001190
       IF(KJRES3 .GT. 5) KP=1                                              001191
       DO 891 K=1,JRES3                                                    001192
       IF(DABS(DBZERO(K,1)) .LE. 1.D-20) DBZERO(K,1)=0.0                   001193
       IF(DABS(DBZERO(K,2)) .LE. 1.D-20) DBZERO(K,2)=0.0                   001194
       DO 890 I=1,KP                                                       001195
       CLPOLE(KK,1)=DBZERO(K,1)                                            001196
       CLPOLE(KK,2)=DBZERO(K,2)                                            001197
C      CLPOLE1(KK,1)=DBZERO(K,1)                                           001198
C      CLPOLE1(KK,2)=DBZERO(K,2)                                           001199
  890 KK=KK+1                                                              001200
  891 CONTINUE                                                             001201
  892 CONTINUE                                                             001202
C                                                                          001203
C  CANCEL EQUAL ZEROS AND POLES VIA THE SUBROUTINE 'CANROOT'               001204
C                                                                          001205
       TOLR=1.D-11                                                         001206
       TOLI=1.D-11                                                         001207
       NZ=NCLZ                                                             001208
       NP=NCLP                                                             001209
       CALL CANROOT(TOLR,TOLI,CLZERO,NZ,CLPOLE,NP)                         001210
```

```
      IF(NZ .EQ. NCLZ) GO TO 950                                        001211
C                                                                       001212
C  OBTAIN NEW NUMERATOR POLYNOMIAL VIA SUBROUTINE 'COMPOLY'             001213
C                                                                       001214
      NCLZ=NZ                                                           001215
      NCLP=NP                                                           001216
C     NCLZ1=NZ                                                          001217
C     NCLP1=NP                                                          001218
      IF(NCLZ .EQ. 0) GO TO 910                                         001219
      DO 905 I=1,NCLZ                                                   001220
      A(I)=CLZERO(I,1)                                                  001221
      B(I)=CLZERO(I,2)                                                  001222
C     CLZERO1(I,1)=CLZERO(I,1)                                          001223
C     CLZERO1(I,2)=CLZERO(I,2)                                          001224
  905 CONTINUE                                                          001225
      CALL COMPOLY(C6,C3,A,B,NCLZ)                                      001226
C                                                                       001227
C  TRANSFER NEW NUMERATOR POLY INTO ARRAY 'RES1N'                       001228
C                                                                       001229
      PWR=NCLZ                                                          001230
      K=1                                                               001231
      NZ=NCLZ+1                                                         001232
      KRES1N=3*NZ                                                       001233
      DO 906 I=1,NZ                                                     001234
      IF(DABS(C6(I)) .LE. 1.D-20) C6(I)=0.0                             001235
      IF(DABS(C3(I)) .LE. 1.D-20) C3(I)=0.0                             001236
      RES1N(K)=C6(I)                                                    001237
      RES1N(K+1)=C3(I)                                                  001238
      RES1N(K+2)=PWR                                                    001239
C     CLNPOL1(I)=C6(I)*(CLNK/CLDK)                                      001240
      PWR=PWR-1.0                                                       001241
  906 K=K+3                                                             001242
      GO TO 911                                                         001243
  910 RES1N(1)=1.0                                                      001244
      RES1N(2)=0.0                                                      001245
      RES1N(3)=0.0                                                      001246
C     CLNPOL1(1)=CLNK/CLDK                                              001247
C                                                                       001248
C  OBTAIN NEW DENOMINATOR POLYNOMIAL VIA SUBROUTINE 'COMPOLY'           001249
C                                                                       001250
  911 DO 915 I=1,NCLP                                                   001251
      A(I)=CLPOLE(I,1)                                                  001252
      B(I)=CLPOLE(I,2)                                                  001253
C     CLPOLE1(I,1)=CLPOLE(I,1)                                          001254
C     CLPOLE1(I,2)=CLPOLE(I,2)                                          001255
  915 CONTINUE                                                          001256
      CALL COMPOLY(C6,C3,A,B,NCLP)                                      001257
C                                                                       001258
C  TRANSFER NEW DENOMINATOR POLY INTO ARRAY 'RES1D'                     001259
C                                                                       001260
      PWR=NCLP                                                          001261
      K=1                                                               001262
      NP=NCLP+1                                                         001263
      KRES1D=3*NP                                                       001264
      DO 920 I=1,NP                                                     001265
```

```
      IF(DABS(C6(I)) .LE. 1.D-20) C6(I)=0.0                              001266
      IF(DABS(C3(I)) .LE. 1.D-20) C3(I)=0.0                              001267
      RES1D(K)=C6(I)                                                     001268
      RES1D(K+1)=C3(I)                                                   001269
      RES1D(K+2)=PWR                                                     001270
C     CLDPOL1(I)=C6(I)                                                   001271
      PWR=PWR-1.0                                                        001272
  920 K=K+3                                                              001273
  950 CONTINUE                                                           001274
C                                                                        001275
C                                                                        001276
C                                                                        001277
C                                                                        001278
C     ***************************************************************  001279
C     ***************  FINAL OUTPUT WRITE SECTION  ******************  001280
C     ***************       FOR LOW-RATE TF        ******************  001281
C     ***************************************************************  001282
C                                                                        001283
      IF(ZFORM .EQ. 2HZ ) TXFORM=2HZ                                     001284
      IF(ZFORM .EQ. 2HZR) TXFORM=2HZR                                    001285
      WRITE(7,707) TXFORM                                                001286
  707 FORMAT(1H1,/////,11X,A2,'-PLANE  LOW-RATE NUMERATOR',/)            001287
      DO 205 I=1,KRES1N,3                                                001288
      IDBN=RES1N(I+2)                                                    001289
  205 WRITE(7,708) RES1N(I),TXFORM,IDBN                                  001290
  708 FORMAT(1H ,10X,'(',G18.10,' )',A2,3H** ,I2)                        001291
      WRITE(7,709) TXFORM                                                001292
  709 FORMAT(1H ,///,10X,A2,'-PLANE  LOW-RATE DENOMINATOR',/)            001293
      DO 206 I=1,KRES1D,3                                                001294
      IDBD=RES1D(I+2)                                                    001295
  206 WRITE(7,708) RES1D(I),TXFORM,IDBD                                  001296
      WRITE(7,710) TXFORM                                                001297
  710 FORMAT(1H ,///,14X,A2,'-PLANE  LOW-RATE ZEROS',/)                  001298
      IF(NCLZ .EQ. 0) GO TO 615                                          001299
      WRITE(7,711) (CLZERO(I,1),CLZERO(I,2),I=1,NCLZ)                    001300
  711 FORMAT(1H ,5X,'(',G18.10,',',G18.10,' )')                          001301
  615 WRITE(7,712) TXFORM                                                001302
  712 FORMAT(1H ,///,14X,A2,'-PLANE  LOW-RATE POLES',/)                  001303
      WRITE(7,711) (CLPOLE(I,1),CLPOLE(I,2),I=1,NCLP)                    001304
      WRITE(7,713) TXFORM,DBCLK                                          001305
  713 FORMAT(1H ,/, 6X,'LOW-RATE ',A2,'-PLANE  GAIN =',G18.10)           001306
      WRITE(7,714) TINT                                                  001307
  714 FORMAT(1H ,5X,'HIGH-RATE SAMPLING PERIOD (SEC) =',G18.10)          001308
      WRITE(7,715) TEXT                                                  001309
  715 FORMAT(1H ,5X,'LOW-RATE  SAMPLING PERIOD (SEC) =',G18.10)          001310
      STOP                                                               001311
      END                                                                001312
C                                                                        001313
C                                                                        001314
C                                                                        001315
C                                                                        001316
C----------------------------------------------------------------------- 001317
CDECK BILIN                                                              001318
      SUBROUTINE BILIN(ALPHA,SIGMA1,SIGMA2,T,KEY,ZFORM,NDPOLE)           001319
C----------------------------------------------------------------------- 001320
```

```
C                                                                       001321
C     THIS SUBROUTINE PERFORMS THE BILINEAR TRANSFORMATION FROM         001322
C     THE Z TO THE W OR W' PLANE OR VISE VERSA.                         001323
C                                                                       001324
C          Z = 1+W/1-W          Z = (2/T)+W'/(2/T)-W'                   001325
C                                                                       001326
C          W = Z-1/Z+1          W' = (2/T)*(Z-1)/(Z+1)                  001327
C                                                                       001328
C     AN EXAMPLE OF HOW THIS TRANSFORMATION IS PERFORMED IS             001329
C     SHOWN BELOW.                                                      001330
C                                                                       001331
C          [(B1*Z)+(B0)]/[(A2*Z**2)+(A1*Z)+(A0)]                        001332
C                                                                       001333
C     [B1*(A+W')*(A-W')]+[B0*(A-W')**2]   ---- NUMERATOR                001334
C     [A2*(A+W')**2]+[A1*(A+W')*(A-W')]+[A0*(A-W')**2]  --  DENOM       001335
C                                                                       001336
C     THE INDIVIDUAL [(A+W')**I]*[(A-W')**(I-1)] TERMS ARE              001337
C     CALCUATED IN SUBROUTINE 'TERM'.                                   001338
C                                                                       001339
C     NOTE:  THE INPUT CLNPOLY AND CLDPOLY ARE IN DESCENDING            001340
C     POWERS WITH ZERO COEFFICIENTS INCLUDED AS PLACE HOLDERS.          001341
C     THE TRANSFORMED POLNOMIALS ARE STORED BACK INTO CLNPOLY           001342
C     AND CLDPOLY IN THE SAME FORMAT.  THE TRANSFORMED ZEROS            001343
C     AND POLES ARE STORED IN ARRAYS CLZERO(50,2) AND                   001344
C     CLPOLE(50,2) WITH THE NUMBER OF ZEROS AND POLES STORED            001345
C     IN VARIABLES 'NCLZ' AND 'NCLP' RESPECTIVELY.                      001346
C                                                                       001347
C     THE GENERAL TRANSFORMATION IS GIVEN BY:                           001348
C                                                                       001349
C          Z = ALPHA*[SIGMA1+W]/[SIGMA2+W]                              001350
C          ALPHA = -1.0                                                 001351
C          SIGMA1 = 1.0                                                 001352
C          SIGMA2 = -1.0                                                001353
C                                                                       001354
C          Z = ALPHA*[SIGMA1+W']/[SIGMA2+W']                            001355
C          ALPHA = -1.0                                                 001356
C          SIGMA1 = 2.0/T                                               001357
C          SIGMA2 = -2.0/T                                              001358
C                                                                       001359
C          W = ALPHA*[SIGMA1+Z]/[SIGMA2+Z]                              001360
C          ALPHA = 1.0                                                  001361
C          SIGMA1 = -1.0                                                001362
C          SIGMA2 = 1.0                                                 001363
C                                                                       001364
C          W' = ALPHA*[SIGMA1+Z]/SIGMA2+Z]                              001365
C          ALPHA = 2.0/T                                                001366
C          SIGMA1 = -1.0                                                001367
C          SIGMA2 = 1.0                                                 001368
C                                                                       001369
C     ADDITIONAL ARGUMENT VARIABLES:                                    001370
C                                                                       001371
C          T = SAMPLING PERIOD (SEC) TO BE USED IN ROOT                 001372
C              TRANSFORMATION IN SUBROUTINE 'SZWROOT'                   001373
C          KEY = INTEGER THAT SPECIFIES PARTICULAR ROOT                 001374
C                TRANSFORMATION IN SUBROUTINE 'SZWROOT'                 001375
```

```
C          NDPOLES = NUMBER OF DISTINCT POLES SENT TO                    001376
C                    SUBROUTINE 'SZWROOT'                                001377
C          ZFORM = ALPHANUMERIC (A2) FLAG THAT INDICATES                 001378
C                  EITHER THE 'Z' OR 'ZR' OPTIONS                        001379
C          NOTE:  COMMON/TXCONV9/ USED TO TRANSFER POLYS                 001380
C                 AND ROOTS TO AND FROM 'BILIN'                          001381
CC                                                                       001382
C                                                                        001383
      DOUBLE ALPHA,SIGMA1,SIGMA2,CLK,CLNK,CLDK,T                         001384
      DOUBLE A(106),B(106),C3(318),C6(318)                               001385
      DOUBLE CLNPOLY(51),CLDPOLY(51),CLZERO(50,2),CLPOLE(50,2)           001386
      COMMON/TXCONV9/CLNPOLY,CLDPOLY,CLZERO,CLPOLE,                      001387
     * NCLZ,NCLP,CLK,CLNK,CLDK                                           001388
      COMMON/TXCONV2/A,B,C3,C6                                           001389
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER                 001390
C                                                                        001391
      IORD=2*(NORDER+3)                                                  001392
      DO 100 I=1,IORD                                                    001393
      A(I)=0.0                                                           001394
  100 B(I)=0.0                                                           001395
      IORD=6*(NORDER+3)                                                  001396
      DO 150 I=1,IORD                                                    001397
  150 C3(I)=0.0                                                          001398
C                                                                        001399
      M=NCLZ+1                                                           001400
      N=NCLP+1                                                           001401
      DO 1 I=1,M                                                         001402
    1 A(M+1-I)=CLNPOLY(M+1-I)                                            001403
      DO 2 I=1,N                                                         001404
    2 B(N+1-I)=CLDPOLY(N+1-I)                                            001405
      IORD=NORDER+1                                                      001406
      DO 175 I=1,IORD                                                    001407
      CLNPOLY(I)=0.0                                                     001408
  175 CLDPOLY(I)=0.0                                                     001409
C                                                                        001410
      DO 50 I=1,N                                                        001411
      CALL TERM(N-I,N-1,ALPHA,SIGMA1,SIGMA2)                             001412
      DO 50 J=1,N                                                        001413
      CLNPOLY(J)=CLNPOLY(J) + A(I)*C3(J)                                 001414
      CLDPOLY(J)=CLDPOLY(J) + B(I)*C3(J)                                 001415
   50 CONTINUE                                                           001416
C                                                                        001417
C                                                                        001418
C  ELIMINATE ZERO LEADING COEFFICIENT IN NUMERATOR POLY                  001419
C  AND SET NUMBER OF ZEROS (NCLZ)                                        001420
C                                                                        001421
      NZ=N                                                               001422
      II=1                                                               001423
   61 DO 63 I=II,NZ                                                      001424
      IF(CLNPOLY(I) .NE. 0.0) GO TO 64                                   001425
      NZ=NZ-1                                                            001426
      IF(NZ .EQ. 0) GO TO 64                                             001427
      DO 62 J=I,NZ                                                       001428
   62 CLNPOLY(J)=CLNPOLY(J+1)                                            001429
      II=I                                                               001430
```

```
      GO TO 61                                                          001431
   63 CONTINUE                                                          001432
   64 NCLZ=NZ-1                                                         001433
C                                                                       001434
C                                                                       001435
C  ELIMINATE ZERO LEADING COEFFICIENT IN DENOMINATOR POLY               001436
C  AND SET NUMBER OF POLES (NCLP)                                       001437
C                                                                       001438
      NP=N                                                              001439
      II=1                                                              001440
   65 DO 67 I=II,NP                                                     001441
      IF(CLDPOLY(I) .NE. 0.0) GO TO 68                                  001442
      NP=NP-1                                                           001443
      IF(NP .EQ. 0) GO TO 68                                            001444
      DO 66 J=I,NP                                                      001445
   66 CLDPOLY(J)=CLDPOLY(J+1)                                           001446
      II=I                                                              001447
      GO TO 65                                                          001448
   67 CONTINUE                                                          001449
   68 NCLP=NP-1                                                         001450
C                                                                       001451
C                                                                       001452
C  DEVIDE NUMERATOR AND DENOMINATOR POLYS BY HIGHEST                    001453
C  COEFFICIENT OF DENOMINATOR AND SET POLY GAIN                         001454
C  VARIABLES - CLK, CLNK, AND CLDK                                      001455
C                                                                       001456
      CLDK=CLDPOLY(1)                                                   001457
      DO 60 I=1,N                                                       001458
      CLNPOLY(I)=CLNPOLY(I)/CLDK                                        001459
   60 CLDPOLY(I)=CLDPOLY(I)/CLDK                                        001460
      CLNK=CLNPOLY(1)                                                   001461
      CLK=CLNPOLY(1)                                                    001462
      CLDK=CLDPOLY(1)                                                   001463
C                                                                       001464
C                                                                       001465
C  OBTAIN NUMERATOR AND DENOMINATOR ROOTS                               001466
C                                                                       001467
      IF(NCLZ .EQ. 0) GO TO 275                                         001468
      DO 200 I=1,NZ                                                     001469
      A(I)=CLNPOLY(I)                                                   001470
  200 B(I)=0.0                                                          001471
      NZ=NCLZ                                                           001472
      CALL ROOTS(A,B,NZ,C3,C6)                                          001473
C  TRANSFER NUMERATOR ROOTS INTO ARRAY CLZERO(50,2)                     001474
      DO 250 I=1,NZ                                                     001475
      CLZERO(I,1)=C3(I)                                                 001476
  250 CLZERO(I,2)=C6(I)                                                 001477
  275 CONTINUE                                                          001478
C                                                                       001479
      IF(NCLP .EQ. 0) GO TO 375                                         001480
      IF(ZFORM .EQ. 2HZ ) GO TO 310                                     001481
C                                                                       001482
C  THIS SECTION TRANSFERS POLES WITH STRAIGHT TRANSFORMATIONS           001483
C  AND DOES NOT FACTOR DENOMINATOR POLY.                                001484
C   THIS SECTION INVOKED WHEN  ZFORM=2HZR.                              001485
```

```
C                                                                       001486
      DO 305 I=1,NDPOLE                                                 001487
      A(I)=CLPOLE(I,1)                                                  001488
  305 B(I)=CLPOLE(I,2)                                                  001489
      CALL SZWROOT(A,B,NDPOLE,T,KEY)                                    001490
      DO 325 I=1,NDPOLE                                                 001491
      CLPOLE(I,1)=A(I)                                                  001492
  325 CLPOLE(I,2)=B(I)                                                  001493
      GO TO 375                                                         001494
  310 CONTINUE                                                          001495
C                                                                       001496
      DO 300 I=1,NP                                                     001497
      A(I)=CLDPOLY(I)                                                   001498
  300 B(I)=0.0                                                          001499
      NP=NCLP                                                           001500
      CALL ROOTS(A,B,NP,C3,C6)                                          001501
C  TRANSFER DENOMINATOR ROOTS INTO ARRAY CLPOLE(50,2)                   001502
      DO 350 I=1,NP                                                     001503
      CLPOLE(I,1)=C3(I)                                                 001504
  350 CLPOLE(I,2)=C6(I)                                                 001505
  375 CONTINUE                                                          001506
C                                                                       001507
      RETURN                                                            001508
      END                                                               001509
C---- ---------------------------------------------------------------- 001510
CDECK TERM                                                              001511
      SUBROUTINE TERM(I,NP,ALPHA,SIGMA1,SIGMA2)                         001512
C     ------------------------------------------------------------------ 001513
C                                                                       001514
C     THIS SUBROUTINE CALCULATES THE INDIVIDUAL TERMS USED IN THE       001515
C     BILINEAR TRANSFORMATION BETWEEN THE Z, W, AND W' PLANES.          001516
C                                                                       001517
C                 [(A+W)**I] * [(A-W)**J]                               001518
C                 J = NP-I                                              001519
C                                                                       001520
C     THE INDIVIDUAL TERMS CALCULATED IN THIS SUBROUTINE ARE            001521
C     PASSED TO THE SUBROUTINE 'BILIN' VIA ARRAY 'C3' IN                001522
C     COMMON/TXCONV2/.                                                  001523
C                                                                       001524
C                                                                       001525
      DOUBLE ALPHA,SIGMA1,SIGMA2,PWR                                    001526
      DOUBLE RR(50),RI(50),POLYA(102),POLYB(102)                        001527
      DOUBLE A(106),B(106),C3(318),C6(318)                              001528
      COMMON/TXCONV2/A,B,C3,C6                                          001529
C                                                                       001530
C                                                                       001531
      J=NP-I                                                            001532
      JJ=J+1                                                            001533
      II=I+1                                                            001534
      IF(I .EQ. 0) GO TO 20                                             001535
      DO 10 K=1,I                                                       001536
      RR(K)= -SIGMA1                                                    001537
   10 RI(K)= 0.0                                                        001538
      CALL COMPOLY(C6,POLYA,RR,RI,I)                                    001539
      PWR=FLOAT(I)                                                      001540
```

```
      KL=1                                                              001541
      DO 11 JL=1,II                                                     001542
      POLYA(KL)=C6(JL)                                                  001543
      POLYA(KL+1)=PWR                                                   001544
      PWR=PWR-1.0                                                       001545
   11 KL=KL+2                                                           001546
      IPOLYA=2*II                                                       001547
      GO TO 25                                                          001548
   20 POLYA(1)=1.0                                                      001549
      POLYA(2)=0.0                                                      001550
      IPOLYA=2                                                          001551
   25 CONTINUE                                                          001552
C                                                                       001553
      IF(J .EQ. 0) GO TO 40                                             001554
      DO 30 K=1,J                                                       001555
      RR(K)= -SIGMA2                                                    001556
   30 RI(K)= 0.0                                                        001557
      CALL COMPOLY(C6,POLYB,RR,RI,J)                                    001558
      PWR=FLOAT(J)                                                      001559
      KL=1                                                              001560
      DO 31 JL=1,JJ                                                     001561
      POLYB(KL)=C6(JL)                                                  001562
      POLYB(KL+1)=PWR                                                   001563
      PWR=PWR-1.0                                                       001564
   31 KL=KL+2                                                           001565
      IPOLYB=2*JJ                                                       001566
      GO TO 45                                                          001567
   40 POLYB(1)=1.0                                                      001568
      POLYB(2)=0.0                                                      001569
      IPOLYB=2                                                          001570
   45 CONTINUE                                                          001571
C                                                                       001572
      CALL MULTIP(POLYA,IPOLYA,POLYB,IPOLYB,C6,N3,2)                    001573
C                                                                       001574
C  ADD MISSING COEFFICIENTS TO C6 POLYNOMIAL                            001575
C                                                                       001576
      CALL COEFF(C6,N3,NP,2)                                            001577
      N=2*(NP+1)                                                        001578
      JKK=1                                                             001579
      DO 50 JK=1,N,2                                                    001580
      C3(JKK)=C6(JK)*(ALPHA**I)                                         001581
   50 JKK=JKK+1                                                         001582
C                                                                       001583
      RETURN                                                            001584
      END                                                               001585
C-----------------------------------------------------------------------001586
CDECK COEFF                                                             001587
      SUBROUTINE COEFF(A,NA,NO,KFORM)                                   001588
C-----------------------------------------------------------------------001589
C                                                                       001590
C     THIS SUBROUTINE ADDS THE MISSING POWER TERMS IN AN                001591
C     POLYNOMIAL BY INSERTING A ZERO COEFFICIENT WITH THE               001592
C     APPROPRIATE POWER AND MOVING THE ORIGINAL TERMS TO                001593
C     MAKE ROOM FOR THIS MISSING TERM.                                  001594
C     A = DOUBLE PRECISION ARRAY CONTAINING INPUT POLY IN               001595
```

```
C          FORMAT SPECIFIED BY KFORM                                      001596
C      NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'                      001597
C      NO = ORDER OF INPUT POLY IN ARRAY 'A'                              001598
C      KFORM = INTEGER SPECIFYING POLY FORMAT                             001599
C      KFORM = 2  ----  (REAL COEFF, PWR)                                 001600
C      KFORM = 3  ----  (REAL COEFF, IMAG COEFF, PWR)                     001601
C      OUTPUT:                                                            001602
C            A = POLY WITH ZERO INSERTED FOR MISSING COEFFS               001603
C            NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'                001604
C                                                                         001605
C                                                                         001606
      DOUBLE A(1),DPWR                                                    001607
      IF(NA .EQ. KFORM) GO TO 50                                          001608
      CALL SIMPLE(A,NA,KFORM)                                             001609
      CALL ORDER3(A,NA,KFORM)                                             001610
C  SET NUMBER OF ELEMENTS ACCORDING TO POLY ORDER                         001611
   50 N=2*(NO+1)                                                          001612
      IF(KFORM .EQ. 3) N=3*(NO+1)                                         001613
      IF(NA .EQ. N) RETURN                                                001614
C  INITIALIZE ADDITIONAL ELEMENTS THAT ARE NEEDED IN THE                  001615
C  POLY ARRAY 'A' TO MAKE ROOM FOR MISSING TERMS.                         001616
      N1=NA+1                                                             001617
      DO 100 I=N1,N                                                       001618
  100 A(I)=0.0                                                            001619
C  INITIALIZE THE POWER COUNTER 'PWR'                                     001620
      PWR=FLOAT(NO)                                                       001621
      DPWR=FLOAT(NO)                                                      001622
      DO 200 I=1,N,KFORM                                                  001623
      JKEY=I+KFORM-1                                                      001624
      R=A(JKEY)                                                           001625
C  CHECK FOR MISSING POWERS                                               001626
      IF(PWR-R) 300,210,300                                               001627
C  SHIFT COEFFICIENTS AND POWERS UP TWO ELEMENTS IN                       001628
C  ARRAY 'A' FROM THE POINT OF THE MISSING POWER AS                       001629
C  SPECIFIED BY JKEY                                                      001630
  300 NX=NA+KFORM-JKEY                                                    001631
      DO 350 J=1,NX                                                       001632
      NN=NA+(KFORM+1)-J                                                   001633
  350 A(NN)=A(NN-KFORM)                                                   001634
      NA=NA+KFORM                                                         001635
      A(JKEY-1)=0.0                                                       001636
      A(JKEY)=DPWR                                                        001637
      IF(KFORM .EQ. 3) A(JKEY-2)=0.0                                      001638
C  REDUCE POWER COUNTER BY ONE (1)                                        001639
  210 DPWR=DPWR-1.0                                                       001640
  200 PWR=PWR-1.0                                                         001641
      NA=N                                                                001642
      RETURN                                                              001643
      END                                                                 001644
C-------------------------------------------------------------------------001645
CDECK WZBILIN                                                             001646
      SUBROUTINE WZBILIN(A,NA,B,NB,D,NDPOLE,NRD,NCD,                      001647
     * ALPHA,SIGMA1,SIGMA2,KFORM,T,KEY,ZFORM)                             001648
C-------------------------------------------------------------------------001649
C                                                                         001650
```

```
C     THIS SUBROUTINE CALCULATES THE SPECIFIC BILINEAR                  001651
C     TRANSFORMATION FROM THE W'-PLANE TO THE Z-PLANE.                  001652
C                                                                       001653
C     A = DOUBLE PRECISION ARRAY CONTAINING INPUT W-PLANE               001654
C         NUMERATOR POLY AND OUTPUT Z-PLANE NUMERATOR POLY              001655
C         IN FORMAT SPECIFIED BY KFORM                                  001656
C     NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'                     001657
C     B = SAME AS ARRAY 'A' FOR DENOMINATOR POLY                        001658
C     NB = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'B'                     001659
C     D = DOUBLE PRECISION ARRAY CONTAINING POLES (REAL,IMAG)           001660
C     NDPOLES = NUMBER OF DISTINCT POLES                                001661
C     NRD = NUMBER OF ROWS IN ARRAY 'D'                                 001662
C     NCD = NUMBER OF COLUMNS IN ARRAY 'D'                              001663
C     NOTE:  FOR DEFINITION OF REMAINING ARGUMENTS SEE                  001664
C            SUBROUTINE 'BILIN' AND 'COEFF'                             001665
C                                                                       001666
C                                                                       001667
      DOUBLE A(1),B(1),D(NRD,NCD),PWR,T                                 001668
      DOUBLE CLK,CLNK,CLDK,ALPHA,SIGMA1,SIGMA2                          001669
      DOUBLE CLNPOLY(51),CLDPOLY(51),CLZERO(50,2),CLPOLE(50,2)          001670
      COMMON/TXCONUS/CLNPOLY,CLDPOLY,CLZERO,CLPOLE,                     001671
     1 NCLZ,NCLP,CLK,CLNK,CLDK                                          001672
C                                                                       001673
C  ORDER POLYS IN DESCENDING POWERS AND ADD MISSING COEFFICIENTS        001674
      CALL ORDER3(A,NA,KFORM)                                           001675
      NOZ=A(KFORM)                                                      001676
      CALL COEFF(A,NA,NOZ,KFORM)                                        001677
      CALL ORDER3(B,NB,KFORM)                                           001678
      NOP=B(KFORM)                                                      001679
      CALL COEFF(B,NB,NOP,KFORM)                                        001680
C                                                                       001681
C  TRANSFER REAL POLY COEFFICIENTS TO ARRAYS CLNPOLY AND CLDPOLY        001682
      K=1                                                               001683
      DO 100 I=1,NA,KFORM                                               001684
      CLNPOLY(K)=A(I)                                                   001685
  100 K=K+1                                                             001686
      K=1                                                               001687
      DO 110 I=1,NB,KFORM                                               001688
      CLDPOLY(K)=B(I)                                                   001689
  110 K=K+1                                                             001690
      NCLZ=NOZ                                                          001691
      NCLP=NOP                                                          001692
C                                                                       001693
C  TRANSFER  POLES INTO ARRAY  CLPOLE                                   001694
      DO 130 I=1,NDPOLE                                                 001695
      CLPOLE(I,1)=D(I,1)                                                001696
  130 CLPOLE(I,2)=D(I,2)                                                001697
C                                                                       001698
C                                                                       001699
C  CALL SUBROUTINE BILIN TO PERFORM THE GENERAL BILINEAR                001700
C  TRANSFORMATION OF THE POLYNOMIALS AND ROOTS                          001701
      CALL BILIN(ALPHA,SIGMA1,SIGMA2,T,KEY,ZFORM,NDPOLE)                001702
      N=NCLZ+1                                                          001703
      M=NCLP+1                                                          001704
C                                                                       001705
```

```
C                                                                       001706
C  TRANSFER NEW NUMINATOR POLY COEFFICIENTS INTO ARRAY 'A'              001707
C  IN THE PROPER FORMAT AS SPECIFIED BY 'KFORM'                         001708
      PWR=FLOAT(NCLZ)                                                   001709
      K=1                                                               001710
      DO 200 I=1,N                                                      001711
      A(K)=CLNPOLY(I)                                                   001712
      A(K+1)=PWR                                                        001713
      IF(KFORM .EQ. 3) A(K+1)=0.0                                       001714
      IF(KFORM .EQ. 3) A(K+2)=PWR                                       001715
      PWR=PWR-1.0                                                       001716
  200 K=K+KFORM                                                         001717
      NA=N*KFORM                                                        001718
C                                                                       001719
C                                                                       001720
C  TRANSFER NEW DENOMINATOR POLY COEFFICIENTS INTO ARRAY 'B'            001721
C  IN THE PROPER FORMAT AS SPECIFIED BY 'KFORM'                         001722
      PWR=FLOAT(NCLP)                                                   001723
      K=1                                                               001724
      DO 300 I=1,M                                                      001725
      B(K)=CLDPOLY(I)                                                   001726
      B(K+1)=PWR                                                        001727
      IF(KFORM .EQ. 3) B(K+1)=0.0                                       001728
      IF(KFORM .EQ. 3) B(K+2)=PWR                                       001729
      PWR=PWR-1.0                                                       001730
  300 K=K+KFORM                                                         001731
      NB=M*KFORM                                                        001732
C                                                                       001733
C  TRANSFER NEW  POLES INTO ARRAY 'D'                                   001734
      KP=NCLP                                                           001735
      IF(ZFORM .EQ. 2HZR) KP=NDPOLE                                     001736
      DO 500 I=1,KP                                                     001737
      D(I,1)=CLPOLE(I,1)                                                001738
  500 D(I,2)=CLPOLE(I,2)                                                001739
      IF(ZFORM .EQ. 2HZR) RETURN                                        001740
      NDPOLE=NCLP                                                       001741
C                                                                       001742
      RETURN                                                            001743
      END                                                               001744
C-----------------------------------------------------------------------001745
CDECK SZWROOT                                                           001746
      SUBROUTINE SZWROOT(RR,RI,NR,T,KEY)                                001747
C-----------------------------------------------------------------------001748
C                                                                       001749
C     THIS SUBROUTINE PERFORMS THE ROOT CONVERSION BETWEEN              001750
C     THE S, Z, W, AND W' PLANES.                                       001751
C                                                                       001752
C     THE DESIRED CONVERSION  IS SELECTED BY THE INPUT                  001753
C     VARIABLE 'KEY':                                                   001754
C                                                                       001755
C     1 = CONVERSION FROM Z TO W PLANE                                  001756
C     2 = CONVERSION FROM Z TO W' PLANE                                 001757
C     3 = CONVERSION FROM W TO Z PLANE                                  001758
C     4 = CONVERSION FROM W' TO Z PLANE                                 001759
C     5 = CONVERSION FROM S TO Z PLANE                                  001760
```

```
C     6 - CONVERSION FROM Z TO S PLANE                                  001761
C     7 - CONVERSION FROM S TO W PLANE                                  001762
C     8 - CONVERSION FROM S TO W' PLANE                                 001763
C     9 - CONVERSION FROM W TO S PLANE                                  001764
C    10 - CONVERSION FROM W' TO S PLANE                                 001765
C                                                                       001766
C     RR - DOUBLE PRECISION ARRAY CONTAINING REAL PART OF ROOT          001767
C     RI - DOUBLE PRECISION ARRAY CONTAINING IMAG PART OF ROOT          001768
C     NR - NUMBER OF ROOTS                                              001769
C      T - SAMPLING PERIOD (SEC)                                        001770
C                                                                       001771
C                                                                       001772
      DOUBLE RR(1),RI(1),T,AP,DEN                                       001773
      GO TO(1,1,2,2,3,4,5,5,6,6),KEY                                    001774
C                                                                       001775
    1 CONTINUE                                                          001776
C                                                                       001777
C  TRANSFORMATION EQUATIONS FOR Z TO W' AND Z TO W:                     001778
C                                                                       001779
C           Z = 1-W/1+W           Z = (2/T)+W'/(2/T)-W'                 001780
C                                                                       001781
C     XW = AP*[(XZ*XZ)+(YZ*YZ)-1]/[(XZ+1)**2+YZ**2]                     001782
C     YW = AP*[2*YZ]/[(XZ+1)**2+YZ**2]                                  001783
C                                                                       001784
C      AP = 1.0  Z TO W          AP = 2/T  Z TO W'                      001785
C                                                                       001786
      AP=1.0                                                            001787
      IF(KEY .EQ. 2) AP=2.0/T                                           001788
      NRR=NR                                                            001789
      DO 10 I=1,NR                                                      001790
   13 CONTINUE                                                          001791
      DEN=(RR(I)+1.)*(RR(I)+1.)+RI(I)*RI(I)                             001792
C                                                                       001793
C  CHECK FOR  Z = -1.0 WHICH TRANSFORMS TO W=W'=INFINITY                001794
      IF(DABS(DEN) .GT. 1.D-12) GO TO 11                                001795
      NRR=NRR-1                                                         001796
      IF(I .GT. NRR) GO TO 10                                           001797
      DO 12 J=I,NRR                                                     001798
      RR(J)=RR(J+1)                                                     001799
   12 RI(J)=RI(J+1)                                                     001800
      WRITE(7,15)                                                       001801
   15 FORMAT(1H ,'ROOT AT  Z = -1.0  TRANSFORMS TO W=W'=INFINITY',/,     001802
     1 ' THIS ROOT HAS BEEN DELETED')                                   001803
      GO TO 13                                                          001804
C                                                                       001805
   11 RR(I)=AP*((RR(I)*RR(I)+RI(I)*RI(I)-1.))/DEN                       001806
      RI(I)=AP*(2.0*RI(I))/DEN                                          001807
   10 CONTINUE                                                          001808
      NR=NRR                                                            001809
      RETURN                                                            001810
C                                                                       001811
    2 CONTINUE                                                          001812
C                                                                       001813
C  TRANSFORMATION EQUATIONS FOR W TO Z AND W' TO Z:                     001814
C                                                                       001815
```

```
C          W = Z-1/Z+1          W' = (2/T)*Z-1/Z+1                          001816
C                                                                           001817
C     XZ = [1-XW*XW-YW*YW]/[(1-XW)**2+YW**2]                                001818
C     YZ = [2.0*YW]/[(1-XW)**2+YW**2]                                       001819
C                                                                           001820
C          FOR W' PLANE -   XW = T/2*XW,   YW = T/2*YW                      001821
C                                                                           001822
C                                                                           001823
      NRR=NR                                                                001824
      DO 20 I=1,NR                                                          001825
   23 CONTINUE                                                              001826
      IF(KEY .EQ. 4) RR(I)=(T/2.)*RR(I)                                     001827
      IF(KEY .EQ. 4) RI(I)=(T/2.)*RI(I)                                     001828
      DEN=(1.-RR(I))*(1.-RR(I))+RI(I)*RI(I)                                 001829
C                                                                           001830
C  CHECK FOR  W = 1.0   OR  W' = 2/T  WHICH TRANSFORM TO Z=INFINITY         001831
      IF(DABS(DEN) .GT. 1.D-12) GO TO 21                                    001832
      NRR=NRR-1                                                             001833
      IF(I .GT. NRR) GO TO 20                                               001834
      DO 22 J=I,NRR                                                         001835
      RR(J)=RR(J+1)                                                         001836
   22 RI(J)=RI(J+1)                                                         001837
      WRITE(7,25)                                                           001838
   25 FORMAT(1H ,*ROOT AT  W=1.0  OR  W'=2/T  TRANSFORM*,                   001839
     1 * TO Z=INFINITY*,/,* THIS ROOT HAS BEEN DELETED*)                    001840
      GO TO 23                                                              001841
   21 RR(I)=(1.-RR(I)*RR(I)-RI(I)*RI(I))/DEN                                001842
      RI(I)=(2.0*RI(I))/DEN                                                 001843
   20 CONTINUE                                                              001844
      NR=NRR                                                                001845
      RETURN                                                                001846
C                                                                           001847
    3 CONTINUE                                                              001848
C                                                                           001849
C  TRANSFORMATION EQUATIONS FOR S TO Z:                                     001850
C                                                                           001851
C     XZ = EXP(XS*T)*COS(YS*T)                                              001852
C     TZ = EXP(XS*T)*SIN(YS*T)                                              001853
C                                                                           001854
C          Z = EXP(S*T)                                                     001855
C                                                                           001856
C                                                                           001857
      DO 30 I=1,NR                                                          001858
      DEN=DEXP(RR(I)*T)                                                     001859
      RR(I)=DEN*DCOS(RI(I)*T)                                               001860
   30 RI(I)=DEN*DSIN(RI(I)*T)                                               001861
      RETURN                                                                001862
C                                                                           001863
    4 CONTINUE                                                              001864
C                                                                           001865
C  TRANSFORMATION EQUATIONS FOR Z TO S:                                     001866
C                                                                           001867
C     XS = (1/2*T)*LN(XZ*XZ+YZ*YZ)                                          001868
C     YS = (1/T)*ARCTAN(YZ/XZ)                                              001869
C                                                                           001870
```

```
C          Z = EXP(S*T)                                                  001871
C                                                                        001872
C                                                                        001873
      DO 40 I=1,NR                                                       001874
      DEN=((1.)/(2.*T))*DLOG(RR(I)*RR(I)+RI(I)*RI(I))                    001875
      RI(I)=(1./T)*DATAN2(RI(I),RR(I))                                   001876
   40 RR(I)=DEN                                                          001877
      RETURN                                                             001878
C                                                                        001879
    5 CONTINUE                                                           001880
C                                                                        001881
C  TRANSFORMATION EQUATIONS FOR S TO W AND S TO W':                      001882
C                                                                        001883
C     DENOMINATOR = EXP(XS*T)+EXP(-XS*T)+2*COS(YS*T)                     001884
C     XW = AP*[EXP(XS*T)-EXP(-XS*T)]/DENOMINATOR                         001885
C     YW = AP*[2*SIN(YS*T)]/DENOMINATOR                                  001886
C                                                                        001887
C                                                                        001888
      AP=1.0                                                             001889
      IF(KEY .EQ. 8) AP=2.0/T                                            001890
      DO 50 I=1,NR                                                       001891
      DEN=DEXP(RR(I)*T)+DEXP(-RR(I)*T)+2.0*DCOS(RI(I)*T)                 001892
      RR(I)=AP*(DEXP(RR(I)*T)-DEXP(-RR(I)*T))/DEN                        001893
   50 RI(I)=AP*(2.0*DSIN(RI(I)*T))/DEN                                   001894
      RETURN                                                             001895
C                                                                        001896
    6 CONTINUE                                                           001897
C                                                                        001898
C  TRANSFORMATION EQUATIONS FOR W TO S AND W' TO S:                      001899
C                                                                        001900
C     XS = (1/2*T)*LN[(1+XW)**2+YW**2]/[(1-XW)**2+YW**2]                 001901
C     YS = (1/T)*ARCTAN[2*YW]/[1-XW*XW-YW*YW]                            001902
C                                                                        001903
C          FOR W' PLANE  -   XW=(T/2)*XW,   YW=(T/2)*YW                  001904
C                                                                        001905
C                                                                        001906
      DO 60 I=1,NR                                                       001907
      IF(KEY .EQ. 10) RR(I)=(T/2.)*RR(I)                                 001908
      IF(KEY .EQ. 10) RI(I)=(T/2.)*RI(I)                                 001909
      DEN=((1.+RR(I))*(1.+RR(I))+RI(I)*RI(I))                            001910
      DEN=DEN/((1.-RR(I))*(1.-RR(I))+RI(I)*RI(I))                        001911
      DEN=(1./(2.*T))*DLOG(DEN)                                          001912
      RI(I)=(1./T)*DATAN2((2.*RI(I)),(1.-RR(I)*RR(I)-RI(I)*RI(I)))       001913
   60 RR(I)=DEN                                                          001914
      RETURN                                                             001915
      END                                                                001916
C-----------------------------------------------------------------------001917
CDECK COMPOLY                                                            001918
      SUBROUTINE COMPOLY(A,B,RR,RI,NR)                                   001919
C-----------------------------------------------------------------------001920
C                                                                        001921
C     THIS SUBROUTINE FORMS THE POLYNOMIAL FROM A SET OF ROOTS.          001922
C     BOTH REAL AND IMAGINARY POLY COEFFICIENTS ARE CALCULATED.          001923
C                                                                        001924
C     A = DOUBLE PRECISION ARRAY CONTAINING REAL COEFFS OF POLY          001925
```

```
C     B = DOUBLE PRECISION ARRAY CONTAINING IMAG COEFFS OF POLY          001926
C     RR = DOUBLE PRECISION ARRAY CONTAINING REAL PART OF ROOT           001927
C     RI = DOUBLE PRECISION ARRAY CONTAINING IMAG PART OF ROOT           001928
C     NR = NUMBER OF ROOTS                                               001929
C                                                                        001930
C                                                                        001931
      DOUBLE A(1),B(1),RR(1),RI(1)                                       001932
      NRR=NR+1                                                           001933
      DO 1 I=2,NRR                                                       001934
      A(I)=0.0                                                           001935
    1 B(I)=0.0                                                           001936
      A(1)=1.0                                                           001937
      B(1)=0.0                                                           001938
      DO 2 I=1,NR                                                        001939
      K=I+1                                                              001940
      DO 2 J=1,I                                                         001941
      A(K)=-A(K-1)*RR(I)+B(K-1)*RI(I)+A(K)                               001942
      B(K)=-B(K-1)*RR(I)-A(K-1)*RI(I)+B(K)                               001943
    2 K=K-1                                                              001944
      RETURN                                                             001945
      END                                                                001946
C                                                                        001947
C------------------------------------------------------------------------001948
CDECK ORDPOLE                                                            001949
      SUBROUTINE ORDPOLE(DBPOLE,IROW,ICOL,NCLPDB,NPOLES,MULPOLE,NM)      001950
C------------------------------------------------------------------------001951
C                                                                        001952
C     THIS SUBROUTINE CHECKS FOR MULTIPLE POLES AND STORES THE           001953
C     MULTIPLICITY IN ARRAY 'MULPOLE'.  THE EXTRA MULTIPLE               001954
C     POLES ARE DELETED AND ONLY A SINGLE COPY OF EACH POLE              001955
C     IS STORED IN ARRAY 'DBPOLE'.                                       001956
C                                                                        001957
C     DBPOLE(IROW,ICOL) = DOUBLE PRECISION ARRAY CONTAINING POLES        001958
C                   (REAL,IMAG)                                          001959
C     MULPOLE = SINGLE PRECISION ARRAY CONTAINING POLE                   001960
C               MULTIPLICITY                                             001961
C     NCLPDB = NUMBER OF TOTAL POLES INPUTTED .                          001962
C     NPOLES = NUMBER OF DISTINCT POLES OUTPUTTED.                       001963
C     NM = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'MULPOLE'                001964
C          ALSO, NUMBER OF DISTINCT POLES OUTPUTTED.                     001965
C     IROW = ROW DIMENSION OF ARRAY 'DBPOLE'                             001966
C     ICOL = COLUMN DIMENSION OF ARRAY 'DBPOLE'                          001967
C                                                                        001968
      DOUBLE DBPOLE(IROW,ICOL),D1,D2                                     001969
      DIMENSION MULPOLE(1)                                               001970
      DO 510 I=1,NCLPDB                                                  001971
  510 MULPOLE(I)=1                                                       001972
      L=1                                                                001973
      NPOLES=NCLPDB                                                      001974
  555 DO 590 I=L,NPOLES                                                  001975
      M=I+1                                                              001976
C  CHECK FOR LAST POLE                                                   001977
      IF(M-NPOLES) 518,518,550                                           001978
  518 CONTINUE                                                           001979
      DO 535 J=M,NPOLES                                                  001980
```

```
C  SET PERCENTAGE TOLERANCE USED TO CHECK POLES                          001981
      D1=DABS(DBPOLE(I,1)*.0001)                                        001982
      D2=DABS(DBPOLE(I,2)*.0001)                                        001983
C  CHECK FOR ZERO IMAGINARY PART OF POLE BEING CHECKED                  001984
C  AGAINST REMAINING POLES.                                             001985
      IF(DBPOLE(I,2)) 519,520,519                                       001986
C  CHECK FOR ZERO REAL PART OF NEXT POLE TO BE CHECKED                  001987
  519 IF(DBPOLE(J,1)) 521,520,521                                       001988
C  SET DECIMAL TOLERANCE FOR REAL PART OF POLE IF IMAG                  001989
C  PART OF POLE BEING CHECKED AGAINST REMAINING POLES                   001990
C  IS ZERO, OR IF REAL PART OF NEXT POLE IS ZERO.                       001991
  520 D1=.00001                                                         001992
  521 CONTINUE                                                          001993
C  CHECK FOR ZERO IMAGINARY PART OF POLE BEING CHECKED                  001994
C  AGAINST REMAINING POLES                                              001995
      IF(DBPOLE(I,2)) 522,523,522                                       001996
C  CHECK FOR ZERO IMAG PART OF NEXT POLE TO BE CHECKED                  001997
  522 IF(DBPOLE(J,2)) 524,523,524                                       001998
C  SET DECIMAL TOLERANCE FOR IMAG PART OF POLE IF IMAG                  001999
C  PART OF POLE BEING CHECKED AGAINST REMAINING POLES                   002000
C  IS ZERO, OR IF IMAG PART OF NEXT POLE IS ZERO.                       002001
  523 D2=.00001                                                         002002
  524 CONTINUE                                                          002003
C  HUNT FOR EQUAL POLES THAT MEET THE 'D1' AND 'D2' TOLERANCE           002004
      IF(DABS(DBPOLE(I,1)-DBPOLE(J,1))-D1) 525,525,535                  002005
  525 IF(DABS(DBPOLE(I,2)-DBPOLE(J,2))-D2) 526,526,560                  002006
C  HUNT FOR POLES WHOSE REAL PARTS MEET THE D1 TOLERANCE                002007
C  AND HAVE AN IMAG PART WHICH IS .LE. 1.D-6 THE REAL                   002008
C  PART.                                                                002009
  560 IF(DBPOLE(I,1)) 561,562,561                                       002010
  562 IF(DABS(DBPOLE(J,2)) .LE. 1.D-6) GO TO 526                        002011
      GO TO 535                                                         002012
  561 IF(DABS(DBPOLE(J,2)/DBPOLE(I,1)) .LE. 1.D-6) GO TO 526            002013
      GO TO 535                                                         002014
  526 CONTINUE                                                          002015
C  EQUAL POLES FOUND, OVERWRITE POLE AND INCREMENT MULTIPLICITY         002016
      NPOLES=NPOLES-1                                                   002017
      MULPOLE(I)=MULPOLE(I)+1                                           002018
      DO 530 K=J,NPOLES                                                 002019
      MULPOLE(K)=MULPOLE(K+1)                                           002020
      DBPOLE(K,1)=DBPOLE(K+1,1)                                         002021
  530 DBPOLE(K,2)=DBPOLE(K+1,2)                                         002022
      L=I                                                               002023
      IF(NPOLES-L-1) 550,555,555                                        002024
  535 CONTINUE                                                          002025
  500 CONTINUE                                                          002026
  550 CONTINUE                                                          002027
      NM=NPOLES                                                         002028
      RETURN                                                            002029
      END                                                               002030
C------------------------------------------------------------------------ 002031
CDECK CANROOT                                                           002032
      SUBROUTINE CANROOT(TOLR,TOLI,CZERO,NZ,CPOLE,NP)                   002033
C------------------------------------------------------------------------ 002034
C                                                                       002035
```

```
C    THIS SUBROUTINE CANCELS EQUAL ZEROS AND POLES ACCORDING TO A          002036
C    SPECIFIED TOLERANCE.  SEPARATE TOLERANCES ARE PROVIDED FOR            002037
C    THE REAL PART OF EACH ROOT (TOLR) AND FOR THE IMAGINARY PART          002038
C    OF EACH ROOT (TOLI).                                                  002039
C                                                                          002040
C           TOLR = TOLERANCE PERCENTAGE FOR REAL PART OF ROOTS             002041
C           TOLI = TOLERANCE PERCENTAGE FOR IMAG PART OF ROOTS             002042
C           CZERO = ARRAY CONTAINING ZEROS (REAL,IMAG)                     002043
C           CPOLE = ARRAY CONTAINING POLES (REAL,IMAG)                     002044
C           NZ= NUMBER OF ZEROS                                            002045
C           NP= NUMBER OF POLES                                            002046
C                                                                          002047
C                                                                          002048
      DOUBLE TOLR,TOLI,CZERO(50,2),CPOLE(50,2),D1,D2                       002049
      L=1                                                                  002050
  150 DO 100 I=L,NZ                                                        002051
      II=I                                                                 002052
      DO 100 J=1,NP                                                        002053
      JJ=J                                                                 002054
      D1=DABS(CZERO(I,1)*TOLR)                                             002055
      D2=DABS(CZERO(I,2)*TOLI)                                             002056
      IF(CZERO(I,1)) 510,511,510                                           002057
  510 IF(CPOLE(J,1)) 512,511,512                                           002058
  511 D1=1.D-11                                                            002059
  512 IF(CZERO(I,2)) 513,514,513                                           002060
  513 IF(CPOLE(I,2)) 515,514,515                                           002061
  514 D2=1.D-11                                                            002062
  515 IF(DABS(CZERO(I,1)-CPOLE(J,1))-D1) 516,516,100                       002063
  516 IF(DABS(CZERO(I,2)-CPOLE(J,2))-D2) 517,517,100                       002064
  517 CONTINUE                                                             002065
      NZ=NZ-1                                                              002066
      NP=NP-1                                                              002067
      DO 200 K=II,NZ                                                       002068
      CZERO(K,1)=CZERO(K+1,1)                                              002069
  200 CZERO(K,2)=CZERO(K+1,2)                                              002070
      DO 300 K=JJ,NP                                                       002071
      CPOLE(K,1)=CPOLE(K+1,1)                                              002072
  300 CPOLE(K,2)=CPOLE(K+1,2)                                              002073
      L=I                                                                  002074
      IF(NZ-L) 400,150,150                                                 002075
  100 CONTINUE                                                             002076
  400 CONTINUE                                                             002077
      RETURN                                                               002078
      END                                                                  002079
C--------------------------------------------------------------------------002080
CDECK DERIV3                                                               002081
      SUBROUTINE DERIV3(P,NP,A,NA,K)                                       002082
C--------------------------------------------------------------------------002083
C                                                                          002084
C   THIS SUBROUTINE TAKES THE DERIVATIVE OF A POLYONOMIAL.                 002085
C                                                                          002086
C          P = INPUT POLY                                                  002087
C         NP = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'P'                    002088
C          A = OUTPUT POLY  ( DERIVATIVE OF POLY 'P' )                     002089
C         NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'                    002090
```

```
C          K = FORMAT OF INPUT POLY                                          002091
C          K = 2  ----  (REAL COEFF, Z-PWR)                                  002092
C          K = 3  ----  (REAL COEFF, IMAG COEFF, Z-PWR)                      002093
C                                                                            002094
C                                                                            002095
      DOUBLE P(1),A(1)                                                       002096
C BRANCH ACCORDING TO INPUT POLY FORMAT                                      002097
      IF(K-2) 10,20,10                                                       002098
C                 K=2  SECTION                                               002099
C  MULTIPLY COEFFICIENTS BY POWERS OF Z                                      002100
C  REDUCE POWER OF Z BY ONE                                                  002101
C  STORE DERIVATIVE IN ARRAY 'A'                                             002102
   20 CONTINUE                                                               002103
      DO 2 I=1,NP,K                                                          002104
      A(I)=P(I+1)*P(I)                                                       002105
      A(I+1)=P(I+1)-1.0                                                      002106
C CHECK FOR NEGATIVE Z PWR AFTER DIFFERENTITAION                             002107
      IF(A(I+1) .LT. 0.0) A(I+1)=0.0                                         002108
    2 CONTINUE                                                               002109
      GO TO 30                                                               002110
C                 K=3  SECTION                                               002111
C  MULTIPLE BOTH REAL & IMAG COEFF BY                                        002112
C  POWER OF Z.                                                               002113
C  REDUCE POWER OF Z BY ONE                                                  002114
C  STORE DERIVATIVE IN ARRAY 'A'                                             002115
   10 CONTINUE                                                               002116
      DO 3 I=1,NP,K                                                          002117
      A(I)=P(I+2)*P(I)                                                       002118
      A(I+1)=P(I+2)*P(I+1)                                                   002119
      A(I+2)=P(I+2)-1.0                                                      002120
C CHECK FOR NEGATIVE Z PWR AFTER DIFFERENIATION                              002121
      IF(A(I+2) .LT. 0.0) A(I+2)=0.0                                         002122
    3 CONTINUE                                                               002123
   30 CONTINUE                                                               002124
C  SET NUMBER OF 'A' ELEMENTS                                                002125
      NA=NP                                                                  002126
      IF(P(NP) .EQ. 0.0) NA=NP-K                                             002127
      RETURN                                                                 002128
      END                                                                    002129
C-------------------------------------------------------------------------   002130
CDECK EVALU3                                                                 002131
      SUBROUTINE EVALU3(P,NP,R,V,ZF,K)                                       002132
C     ---------------------------------------------------------------------  002133
C                                                                            002134
C   THIS SUBROUTINE EVALUATES A POLYONOMIAL FOR A COMPLEX                    002135
C   VALUE OF  (Z = X + JY).  IT CAN EVALUATE BOTH REAL                       002136
C   COEFFICIENT POLYS AND COMPLEX COEFFICIENT POLYS.                         002137
C                                                                            002138
C         P = INPUT POLY TO BE EVALUATED                                     002139
C        NP = NUMBER OF WORKING ELEMENTS IN  P(NP)                           002140
C         R = POLE OR ROOT TO BE EVALUATED  (R=X+JY)                         002141
C         V = VALUE OF POLY 'P' AT POLE 'R' (V=A+JB)                         002142
C        ZF = SCALE FACTOR IMPOSED ON LARGE VALUES OF 'V'                    002143
C             TO MAINTAIN NUMERICAL ACCURACY.                                002144
C         K = FORMAT OF INPUT POLY 'P'                                       002145
```

```
C          K = 2   ----  (REAL COEFF, Z-PWR)                              002146
C          K = 3   ----  (REAL COEFF, IMAG COEFF, Z-PWR)                  002147
C                                                                         002148
C    THE FOLLOWING STEPS REPRESENT AN EXAMPLE OF THE                      002149
C    ALGORITHM USED TO EVALUATE THE INPUT POLY 'P'                        002150
C    AT ROOT 'R':                                                         002151
C                                                                         002152
C          A3*S3 + A2*S2 + A1*S + A0      [ (NP=8)  & (K=2) ]            002153
C          R = S = [R(1) + JR(2)]                                         002154
C      I=1                                                                002155
C      [A3*R(1) + J A3*R(2)]*S2 + A2*S2 + A1*S + A0                       002156
C      [(RP) + J (CP)]*S2 + A2*S2 +A1*S + A0                              002157
C      [(RP+A2) + J (CP)]*S2 + A1*S + A0                                  002158
C      [V(1) + J V(2)]*S2 + A1*S + A0                                     002159
C      I=3                                                                002160
C      [      R ]*S + A1*S + A0                                           002161
C      [      + J (CP)]*S + A1*S + A0                                     002162
C      [(RP+A1) + J (CP)]*S + A0                                          002163
C      [V(1) + J V(2)]*S + A0                                             002164
C      I=5                                                                002165
C      [ V      + A0                                                      002166
C      [(RP)    (CP)] + A0                                                002167
C      [(RP+A0) + J (CP)]                                                 002168
C      [V(1) + J V(2)]                                                    002169
C      I=7        RETURN  AS RESULT OF  IF(P(N)) 539,4,539                002170
C                                                                         002171
C                                                                         002172
C          A3*S3 + A2*S2 + A1*S        [  (NP=6) & (K=2)  ]               002173
C          R = S = [R(1) + J R(2)]                                        002174
C      I=1                                                                002175
C      [A3*R(1) + J A3*R(2)]*S2 + A2*S2 + A1*S                            002176
C      [(RP) + J (CP)]*S2 + A2*S2 + A1*S                                  002177
C      [(RP+A2) + J (CP)]*S2 + A1*S                                       002178
C      [V(1) + J V(2)]*S2 + A1*S                                          002179
C      I=3                                                                002180
C      [ V * R ]*S + A1*S                                                 002181
C      [(RP) + J (CP)]*S + A1*S                                           002182
C      [(RP+A1) + J (CP)]*S                                               002183
C      [V(1) + J V(2)]*S                                                  002184
C      I=5                                                                002185
C      [ V * R ]                                                          002186
C      [(RP) + J (CP)]                                                    002187
C      [V(1) + J (CP)]                                                    002188
C      I=7        -------     RETURN   (NP=6)                             002189
C                                                                         002190
C                                                                         002191
C                                                                         002192
      DOUBLE R(1),EXP,RP,CP,ZF,P(1),V(1)                                  002193
      ZF=1.0                                                              002194
C  ORDER POLY 'P' IN DESCENDING POWERS OF Z                               002195
      CALL ORDER3(P,NP,K)                                                 002196
C  BRANCH ACCORDING TO POLY 'P' FORMAT                                    002197
      IF(K-2) 10,20,10                                                    002198
C              K=2  SECTION                                               002199
C  SET FIRST 2 UNUSED ELEMENTS IN 'P' TO ZERO                             002200
```

```
C  INITIALIZE EXPONENT VARIABLE 'EXP' TO FIRST                          002201
C  Z POWER IN POLY 'P'  AND SET EVAUATION                              002202
C  VARAIABLE 'V' TO FIRST COEFFICIENT OF 'P'                           002203
   20 CONTINUE                                                         002204
      P(NP+1)=0.0                                                      002205
      P(NP+2)=0.0                                                      002206
      EXP=P(2)                                                         002207
      V(1)=P(1)                                                        002208
      V(2)=0.0                                                         002209
      GO TO 25                                                         002210
C              K=3  SECTION                                            002211
C  SET FIRST 3 UNUSED ELEMENTS IN 'P' TO ZERO                          002212
C  INITIALIZE EXPONENT VARIABLE 'EXP' TO FIRST                         002213
C  Z POWER IN POLY 'P'  AND SET EVALUATION                             002214
C  VARIABLE 'V' TO FIRST COEFFICIENT OF 'P'                            002215
   10 CONTINUE                                                         002216
      P(NP+1)=0.0                                                      002217
      P(NP+2)=0.0                                                      002218
      P(NP+3)=0.0                                                      002219
      EXP=P(3)                                                         002220
      V(1)=P(1)                                                        002221
      V(2)=P(2)                                                        002222
   25 CONTINUE                                                         002223
C  CHECK FOR 0TH-ORDER POLY 'P'  ---  RETURN                           002224
      IF(EXP) 538,4,538                                                002225
  538 CONTINUE                                                         002226
C  MASTER DO LOOP FOR EVALUATING POLY 'P'                              002227
      DO 3 I=1,NP,K                                                    002228
C  SET 'N' TO LOCATION OF Z PWR IN POLY 'P'                            002229
      N=I+K-1                                                          002230
C  CKECK FOR ZERO Z PWR IN POLY  'P'  -- RETURN                        002231
C  IF LAST TERM IN POLY 'P' HAS  ZERO Z-PWR,                           002232
C  LAST COEFFICIENT ALREADY ADDED IN LAST LOOP                         002233
C  THROUGH ROUTINE BY V(1) & V(2) CODE BELOW                           002234
C  IF LAST TERM NOT ZERO Z-PWR, ONE MORE                               002235
C  MULTIPLICATION BY ROOT 'R' REQUIRED AND THUS                        002236
C ONE MORE LOOP REQUIRED THROUGH ROUTINE                              002237
      IF(P(N)) 539,4,539                                               002238
  539 CONTINUE                                                         002239
C  MULTIPLY COMPLEX COEFF STORED IN V(1),V(2)                          002240
C  BY COMPLEX POLE IN  R(1),R(2)                                       002241
C  STORE RESULTS IN VARAIABLES  'RP' AND 'CP'                          002242
    1 RP=V(1)*R(1) - V(2)*R(2)                                         002243
      CP=V(1)*R(2) + V(2)*R(1)                                         002244
C  CHECK FOR EXCESSIVELY LARGE VALUES OF 'RP'                          002245
      IF(DABS(RP)-1.D20) 11,540,540                                    002246
  540 CONTINUE                                                         002247
C  SCALE DOUN LARGE VALUES OF EVALATION RESULTS                        002248
C IN VARIABLES 'RP' AND 'CP' BY 1.E10 AND STORE                       002249
C TOTAL SCALE FACTOR IN 'ZF'                                          002250
      ZF=ZF*1.D10                                                      002251
      RP=RP/1.D10                                                      002252
      CP=CP/1.D10                                                      002253
   11 CONTINUE                                                         002254
C  REDUCE  'Z' POWER VARIABLE  BY ONE                                  002255
```

```
      EXP=EXP-1.0                                                       002256
C  SET "NN" TO NEXT POWER OF "Z" IN POLY "P"                           002257
      NN=I+(2*K)-1                                                      002258
C  CHECK FOR MISSING POWER OF "Z" IN POLY "P"                           002259
      IF(EXP-P(NN)) 2,2,541                                             002260
C  BRANCH HERE FOR MISSING Z-POWER                                      002261
C  NOTE:  NEXT COEFF NOT ADDED TO RUNNING SUM                           002262
C  VARIABLE V(1) & J V(2)                                               002263
  541 CONTINUE                                                          002264
      V(1)=RP                                                           002265
      V(2)=CP                                                           002266
      GO TO 1                                                           002267
C BRANCH HERE FOR CONSEQUETIVE Z POWERS                                 002268
C NEXT COEFFICIENT ADDED TO RUNNING                                     002269
C SUM VARIABLE V(1) & J V(2)                                            002270
    2 V(1)=RP+P(I+K)/ZF                                                 002271
C  BRANCH ACCORDING TO POLY "P" FORMAT                                  002272
      IF(K-2) 700,5,700                                                 002273
C  ADD IMAG PART OF NEXT COEFF TO RUNNING                               002274
C SUM VARIABLE V(2)  FOR    K=3                                         002275
  700 V(2)=CP+P(I+K+1)/ZF                                               002276
      GO TO 3                                                           002277
C  SINCE NO IMAGINARY PART OF COFF FOR K=2                              002278
C  SIMPLE TRANSFER PRIOR RUNNING SUM TO V(2)                            002279
    5 V(2)=CP                                                           002280
    3 CONTINUE                                                          002281
    4 RETURN                                                            002282
      END                                                               002283
C-------------------------------------------------------------------------- 002284
CDECK ORDER3                                                            002285
      SUBROUTINE ORDER3(P,N,K)                                          002286
C-------------------------------------------------------------------------- 002287
C                                                                       002288
C     THIS SUBROUTINE ORDERS A POLYNOMIAL IN DESCENDING POWERS          002289
C                                                                       002290
C     P = INPUT POLY                                                    002291
C     N = NUMBER OCCUPIED ELEMENTS IN "P"                               002292
C     K = FORMAT OF INPUT POLY                                          002293
C     K = 2  ----  (REAL COEFF, PWR)                                    002294
C     K = 3  ----  (REAL COEFF, IMAG COEFF, PWR)                        002295
C                                                                       002296
      DOUBLE P(1),TEMP                                                  002297
      M=N-K                                                             002298
      IF(M)5,5,1                                                        002299
    1 REVERS=1.                                                         002300
      DO 4 I=K,M,K                                                      002301
      J=I+K                                                             002302
      IF(P(I)-P(J))2,4,4                                                002303
    2 REVERS=0                                                          002304
      DO 3 L=1,K                                                        002305
      LJ=J-L+1                                                          002306
      LI=I-L+1                                                          002307
      TEMP=P(LJ)                                                        002308
      P(LJ)=P(LI)                                                       002309
      P(LI)=TEMP                                                        002310
```

```
    3 CONTINUE                                                          002311
    4 CONTINUE                                                          002312
      IF(REVERS)5,1,5                                                   002313
    5 RETURN                                                            002314
      END                                                               002315
C-----------------------------------------------------------------------002316
CDECK DIVI                                                              002317
      SUBROUTINE DIVI(A,B,C,D,X,Y)                                      002318
C-----------------------------------------------------------------------002319
C     THIS ROUTINE DEVIDES TWO COMPLEX NUMBERS                          002320
C          X+JY = (A+JB)/(C+JD)                                         002321
C                                                                       002322
      DOUBLE A,B,C,D,X,Y                                                002323
      X=(A*C+B*D)/(C*C+D*D)                                             002324
      Y=(B*C-A*D)/(C*C+D*D)                                             002325
      RETURN                                                            002326
      END                                                               002327
C-----------------------------------------------------------------------002328
CDECK MULT                                                              002329
      SUBROUTINE MULT(A,B,C,D,X,Y)                                      002330
C-----------------------------------------------------------------------002331
C     THIS ROUTINE MULTIPLIES TWO COMPLEX NUMBERS                       002332
C          X+JY = (A+JB)*(C+JD)                                         002333
C                                                                       002334
      DOUBLE A,B,C,D,X,Y                                                002335
      X=A*C-B*D                                                         002336
      Y=B*C+A*D                                                         002337
      RETURN                                                            002338
      END                                                               002339
C-----------------------------------------------------------------------002340
CDECK MULTIP                                                            002341
      SUBROUTINE MULTIP(C1,NT1,C2,NT2,C3,NT3,N)                         002342
C-----------------------------------------------------------------------002343
C                                                                       002344
C     THIS ROUTINE MULTIPLIES TWO POLYNOMIALS                           002345
C          C1 = INPUT POLY                                              002346
C          NT1 = NUMBER OCCUPIED ELEMENTS IN ARRAY 'C1'                 A02347
C          C2 = INPUT POLY                                              002348
C          NT2 = NUMBER OCCUPIED ELEMENTS IN ARRAY 'C2'                 002349
C          C3 = C1*C2                                                   002350
C          NT3 = NUMBER OCCUPIED ELEMENTS IN ARRAY 'C3'                 002351
C          N = FORMAT OF POLYNOMIALS                                    002352
C          N = 2  ----  (REAL COEFF, PWR)                               002353
C          N = 3  ----  (REAL COEFF, IMAG COEFF, PWR)                   002354
C                                                                       002355
      DOUBLE    C1(1),C2(1),C3(1)                                       002356
      K=1-N                                                             002357
      DO 110 I=1,NT1,N                                                  002358
      DO 100 J=1,NT2,N                                                  002359
      K=K+N                                                             002360
      IF (N-3) 529,90,529                                               002361
  529 CONTINUE                                                          002362
      C3(K)=C1(I)*C2(J)                                                 002363
      C3(K+1)=C1(I+1)+C2(J+1)                                           002364
      GO TO 100                                                         002365
```

```
   90  C3(K)   =C1(I)*C2(J)-C1(I+1)*C2(J+1)                                  002366
       C3(K+1)=C1(I)*C2(J+1)+C1(I+1)*C2(J)                                   002367
       C3(K+2)=C1(I+2)+C2(J+2)                                               002368
  100  CONTINUE                                                              002369
       NT3=K+N-1                                                             002370
       IF(I-2) 110,530,530                                                   002371
  530  CONTINUE                                                              002372
       CALL SIMPLE(C3,NT3,N)                                                 002373
       K=NT3-N+1                                                             002374
  110  CONTINUE                                                              002375
       RETURN                                                                002376
       END                                                                   002377
C------------------------------------------------------------------------- 002378
CDECK ADD                                                                   002379
       SUBROUTINE ADD(C1,NT1,C2,NT2,C3,NT3,M)                                002380
C------------------------------------------------------------------------- 002381
C                                                                           002382
C      THIS ROUTINE ADDS TWO POLYNOMIALS                                    002383
C           C1 = INPUT POLY                                                 002384
C           NT1 = NUMBER OCCUPIED ELEMENTS IN ARRAY 'C1'                    002385
C           C2 = INPUT POLY                                                 002386
C           NT2 = NUMBER OCCUPIED ELEMENTS IN ARRAY 'C2'                    002387
C           C3 = C1+C2                                                      002388
C           NT3 = NUMBER OCCUPIED ELEMENTS IN ARRAY 'C3'                    002389
C           M = FORMAT OF POLYNOMIALS                                       002390
C           M = 2  ----  (REAL COEFF, PWR)                                  002391
C           M = 3  ----  (REAL COEFF, IMAG COEFF, PWR)                      002392
C                                                                           002393
       DOUBLE C1(1),C2(1),C3(1)                                             002394
       DO 100 I=1,NT1                                                       002395
  100  C3(I)=C1(I)                                                          002396
       DO 110 I=1,NT2                                                       002397
       J=I+NT1                                                              002398
  110  C3(J)=C2(I)                                                          002399
       NT3=NT1+NT2                                                          002400
       CALL SIMPLE(C3,NT3,M)                                                002401
       RETURN                                                               002402
       END                                                                  002403
C------------------------------------------------------------------------- 002404
CDECK SIMPLE                                                                002405
       SUBROUTINE SIMPLE(C,NT,M)                                            002406
C------------------------------------------------------------------------- 002407
C                                                                           002408
C      THIS ROUTINE SIMPLIFIES A POLYNOMIAL BY ADDING                       002409
C      COEFFICIENTS OF LIKE POWERS                                          002410
C           C = INPUT POLY AND SIMPLIFIED OUTPUT POLY                       002411
C           NT = NUMBER OCCUPIED ELEMENTS IN ARRAY 'C'                      002412
C           M = FORMAT OF POLYNOMIALS                                       002413
C           M = 2  ----  (REAL COEFF, PWR)                                  002414
C           M = 3  ----  (REAL COEFF, IMAG COEFF, PWR)                      002415
C                                                                           002416
       DOUBLE C(2)                                                          002417
       N=NT-M                                                               002418
       IF(N) 155,155,10                                                     002419
   10  DO 110 I=1,N,M                                                       002420
```

```
      JJ=I+M                                                            002421
      DO 110 J=JJ,NT,M                                                  002422
      K=M+I-1                                                           002423
      L=M+J-1                                                           002424
      IF(C(K)-C(L))110,100,110                                          002425
  100 C(I)=C(I)+C(J)                                                    002426
      C(J)=0                                                            002427
      IF(M-2)534,110,534                                                002428
  534 CONTINUE                                                          002429
      C(I+1)=C(I+1)+C(J+1)                                              002430
      C(J+1)=0                                                          002431
  110 CONTINUE                                                          002432
      II=1                                                              002433
  115 DO 140 I=II,NT,M                                                  002434
      IF(C(I))140,535,140                                               002435
  535 CONTINUE                                                          002436
      IF(M-2)536,120,536                                                002437
  536 CONTINUE                                                          002438
      IF(C(I+1))140,537,140                                             002439
  537 CONTINUE                                                          002440
  120 JJ=I+M                                                            002441
      DO 130 J=JJ,NT                                                    002442
      K=J-M                                                             002443
  130 C(K)=C(J)                                                         002444
      NT=NT-M                                                           002445
      II=I                                                              002446
      IF(I+M-1-NT)115,115,145                                           002447
  140 CONTINUE                                                          002448
  145 IF(NT)155,150,155                                                 002449
  150 NT=M                                                              002450
      C(1)=0                                                            002451
      C(2)=0                                                            002452
      C(3)=0                                                            002453
  155 RETURN                                                            002454
      END                                                               002455
C-----------------------------------------------------------------------002456
CDECK ROOTS                                                             002457
      SUBROUTINE ROOTS(A,B,NN,RR,RI)                                    002458
C-----------------------------------------------------------------------002459
C                                                                       002460
C     THIS ROUTINE FINDS THE ROOTS OF POLYNOMIAL                        002461
C          A = REAL COEFF OF POLY IN DESCENDING ORDER                   002462
C          B = IMAG COEFF OF POLY IN DESCENDING ORDER                   002463
C          RR = REAL PART OF ROOTS                                      002464
C          RI = IMAG PART OF ROOTS                                      002465
C          NN = ORDER OF POLYNOMIAL                                     002466
C          [A(1)+JB(1)]*S**(NN) + [A(2)+JB(2)]*S**(NN-1) + ...          002467
C                                                                       002468
      DOUBLE A(1),B(1),RR(1),RI(1)                                      002469
      DOUBLE PRECISION C,D,C1,C2,D1,D2,X,Y,F,G,F1,F2,G1,G2,FH,          002470
     1FK,FM                                                             002471
      L=1                                                               002472
      N=NN                                                              002473
      FM=1.                                                             002474
      NPLUS=N+1                                                         002475
```

```
      DO 265 I=1,NPLUS                                                  002476
      A1=A(I)                                                           002477
      B1=B(I)                                                           002478
      IF(ABS(A1).GT..00000001.OR.ABS(B1).GT..00000001)GO TO 266         002479
      DO 264 J=1,NPLUS                                                  002480
      A(J)=A(J+1)                                                       002481
  264 B(J)=B(J+1)                                                       002482
      N=N-1                                                             002483
      IF(N.EQ.0)RR(1)=0.                                                002484
      IF(N.EQ.0)RI(1)=0.                                                002485
      IF(N.EQ.0)RETURN                                                  002486
  265 CONTINUE                                                          002487
  266 CONTINUE                                                          002488
      NN=N                                                              002489
      DO 303 I=1,NN                                                     002490
      RR(I)=0.                                                          002491
  303 RI(I)=0.                                                          002492
  112 N1=N+1                                                            002493
      IF(N.EQ.1)GO TO 103                                               002494
      D1=DMAX1(DABS(A(N+1)),DABS(B(N+1)))                               002495
      IF(D1.EQ.0.)GO TO 111                                             002496
      D2=DMAX1(DABS(A),DABS(B))                                         002497
      FK=N                                                              002498
      FM=D2**(1.  /FK)/D1**(1.  /FK)                                    002499
      K=N                                                               002500
      DO 3 I=1,N                                                        002501
      A(I)=A(I)/(D1*FM**K)                                              002502
      B(I)=B(I)/(D1*FM**K)                                              002503
    3 K=K-1                                                             002504
      A(N1)=A(N1)/D1                                                    002505
      B(N1)=B(N1)/D1                                                    002506
    7 X=.9876532                                                        002507
      Y=.9654312                                                        002508
      LL=1                                                              002509
      IL=1                                                              002510
    8 C=A                                                               002511
      D=B                                                               002512
      C1=A                                                              002513
      C2=B                                                              002514
      DO 5 I=1,N                                                        002515
      F=X*C-Y*D+A(I+1)                                                  002516
      G=X*D+Y*C+B(I+1)                                                  002517
      IF(I.EQ.N)GO TO 5                                                 002518
      F1=X*C1-Y*C2+F                                                    002519
      F2=X*C2+Y*C1+G                                                    002520
      C=F                                                               002521
      D=G                                                               002522
      C1=F1                                                             002523
    5 C2=F2                                                             002524
      D=F1**2+F2**2                                                     002525
      FH=-(F*F1+G*F2)/D                                                 002526
      FK=-(G*F1-F*F2)/D                                                 002527
   12 X=X+FH                                                            002528
      Y=Y+FK                                                            002529
      IF((X**2+Y**2).EQ.((X+FH)**2+(Y+FK)**2))GO TO 22                  002530
```

```
      IF(X.EQ.0..OR.Y.EQ.0.)GO TO 22                                    002531
      A1=FH/X                                                           002532
      A2=FK/Y                                                           002533
      IF(ABS(A1   ).GT.1.E-4.OR. ABS(A2   ).GT.1.E-4)GO TO 21           002534
      GO TO(211,21),IL                                                  002535
  211 LL=194                                                            002536
      IL=2                                                              002537
   21 LL=LL+1                                                           002538
      IF(LL.GT.200)GO TO 22                                             002539
      GO TO 8                                                           002540
   22 RR(L)=X/FM                                                        002541
      RI(L)=Y/FM                                                        002542
      DO 100 I=1,M                                                      002543
      A(I+1)=X*A(I)-Y*B(I)+A(I+1)                                       002544
  100 B(I+1)=X*B(I)+Y*A(I)+B(I+1)                                       002545
  102 N=N-1                                                             002546
      L=L+1                                                             002547
      IF(N.GT.1)GO TO 7                                                 002548
  103 D=A**2+B**2                                                       002549
      RR(L)=(-A*A(2)-B*B(2))/(D*FM)                                     002550
      RI(L)=(-A*B(2)+B*A(2))/(D*FM)                                     002551
      DO 104 I=1,NN                                                     002552
      X=RR(I)**2                                                        002553
      Y=RI(I)**2                                                        002554
      D=X+Y                                                             002555
      IF(D.EQ.X)RI(I)=0.                                                002556
      IF(D.EQ.Y)RR(I)=0.                                                002557
  104 CONTINUE                                                          002558
      RETURN                                                            002559
  111 RR(L)=0.                                                          002560
      RI(L)=0.                                                          002561
      N=N-1                                                             002562
      L=L+1                                                             002563
      GO TO 112                                                         002564
      END                                                               002565
C-------------------------------------------------------------------- 002566
CDECK DOLOOP                                                            002567
      SUBROUTINE DOLOOP(C,NC,D,ND)                                      002568
C-------------------------------------------------------------------- 002569
C                                                                       002570
C     THIS ROUTINE IMPLEMENTS A STANDARD DO LOOP                        002571
C     TO TRANSFER ONE ARRAY INTO ANOTHER ARRAY                          002572
C          C = INPUT ARRAY                                              002573
C          NC = NUMBER OCCUPIED ELEMENTS IN 'C'                         002574
C          D = OUTPUT ARRAY                                             002575
C          ND = NC                                                      002576
C                                                                       002577
      DOUBLE C(1),D(1)                                                  002578
      ND=NC                                                             002579
      DO 1 I=1,NC                                                       002580
    1 D(I)=C(I)                                                         002581
      RETURN                                                            002582
      END                                                               002583
C                                                                       002584
C-------------------------------------------------------------------- 002585
```

```
CDECK POLE                                                              002586
      SUBROUTINE POLE(REAL,RIMAG)                                       002587
C-----------------------------------------------------------------------002588
C                                                                       002589
C    THIS SUBROUTINE CALULATES A LOW-RATE POLE [ Z=EXP(S*T) ]           002590
C    FROM A GIVEN HIGH-RATE POLE  [ Z=EXP(S*T/N) ].                     002591
C    THE NEW POLE LOCATION IS SIMPLY THE ORIGINAL POLE                  002592
C    LOCATION IN THE Z-DOMAIN TAKENED TO A POWER EQUAL                  002593
C    TO THE RATIO OF THE LOW-RATE TO HIGH-RATE SAMPLING                 002594
C    INTERVALS, XNT = (T) / (T/N).                                      002595
C                                                                       002596
C          [ Z=EXP(S*T) ]          [ Z=EXP(S*T/N) ]                    002597
C           [ NEW POLE ]     =     [ ORIGINAL POLE ]**XNT               002598
C                                                                       002599
C          REAL = REAL PART OF HIGH-RATE POLE ON INPUT                  002600
C                 REAL PART OF LOW-RATE POLE ON OUTPUT                  002601
C          RIMAG = IMAG PART OF HIGH-RATE POLE ON INPUT                 002602
C                  IMAG PART OF LOW-RATE POLE ON OUTPUT                 002603
C          XNT = (T) / (T/N)                                            002604
C                                                                       002605
C                                                                       002606
      DOUBLE REAL,RIMAG,REAL1,REAL2,RIMAG1,C,ANG,X,Y                    002607
      DOUBLE TINT,TEXT,XNT,DBCLK                                        002608
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK         002609
C  EXIT ROUTINE IF XNT=1                                                002610
      IF(XNT .EQ. 1.0) RETURN                                           002611
C  TRANSFER INTEGER VALUE OF XNT INTO NT                                002612
      NT=XNT                                                            002613
C  CHECK FOR INTEGER RATIO XNT                                          002614
      IF(XNT-NT) 50,100,50                                              002615
C  POLE TXFORM SECTION FOR NON-INTEGER XNT                              002616
   50 CONTINUE                                                          002617
C  CHECK FOR SPECIAL CASE OF REAL NEGATIVE POLE                         002618
C  THIS CONDITION EXIST ONLY WHEN  IMAG PART OF                         002619
C  COMPLEX  S-PLANE POLE IS AT EXACTLY ONE-HALF                         002620
C  THE SAMPLING FREQUENCY   W=2*PIE/T.                                  002621
C  FOR THIS CONDITION, THE IMAG PART OF THE                             002622
C  Z-PLANE POLE IS EXACTLY ZERO AND THUS THE                            002623
C  COMPLEX S-PLANE POLE BECOMES A NEGATIVE REAL                         002624
C  POLE IN THE Z-PLANE.                                                 002625
C  THE GENERAL TRANSFORMATION FOR THIS SPECIAL                          002626
C  CASE IS BASED ON THE FOLLOWING:                                      002627
C   Z / Z - [EXP(-XT) * EXP(-JYT)]                                      002628
C   WHERE,  (S + X + JY) = S-PLANE POLE                                 002629
C   Y = 1/T * PIE                                                       002630
C   X = -(1/T) * LN [REAL/COS(Y*T)]                                     002631
C   WITH,   REAL=EXP(-X*T)*COS(Y*T)                                     002632
C         RIMAG= -EXP(-X*T)*SIN(Y*T)                                    002633
C          COS(Y*T) = COS(PIE) = -1                                     002634
C   NOTE:  EXP(-X*T)  IS ALWAYS NEGATIVE                                002635
C                                                                       002636
      IF(REAL .LT. 0.0 .AND. RIMAG .EQ. 0.0) GO TO 51                   002637
      GO TO 52                                                          002638
   51 Y = (1.0/TINT)*(3.141592654)                                      002639
      X = (-1.0/TINT) * DLOG(-REAL)                                     002640
```

```
      REAL = DEXP(-X*TEXT) *  DCOS(Y*TEXT)                              002641
      RIMAG = -DEXP(-X*TEXT) * DSIN(Y*TEXT)                             002642
      GO TO 200                                                         002643
   52 CONTINUE                                                          002644
C                                                                       002645
C  CALCULATE MAGNITUDE OF LOW-RATE POLE                                 002646
      IF(REAL .EQ. 0.0 .AND. RIMAG .EQ. 0.0) GO TO 200                  002647
      C=(REAL*REAL+RIMAG*RIMAG)**(XNT*0.5)                              002648
C  CALCULATE PHASE ANGLE OF LOW-RATE POLE                               002649
      ANG=(DATAN2(RIMAG,REAL))*XNT                                      002650
C  CALCULATE REAL & IMAG PART OF LOW-RATE POLE                          002651
      REAL=C*DCOS(ANG)                                                  002652
      RIMAG=C*DSIN(ANG)                                                 002653
      GO TO 200                                                         002654
C  POLE TXFORM SECTION FOR INTEGER XNT                                  002655
  100 CONTINUE                                                          002656
      REAL1=REAL                                                        002657
      RIMAG1=RIMAG                                                      002658
C  MULTIPLY HIGH-RATE POLE Z=EXP(S*T/N)                                 002659
C  TIMES ITSELF XNT TIMES                                               002660
      NTT=NT-1                                                          002661
      DO 300 K=1,NTT                                                    002662
      REAL2=REAL                                                        002663
      REAL=REAL*REAL1 - RIMAG*RIMAG1                                    002664
      RIMAG=REAL2*RIMAG1 + REAL1*RIMAG                                  002665
  300 CONTINUE                                                          002666
  200 CONTINUE                                                          002667
      RETURN                                                            002668
      END                                                               002669
C                                                                       002670
C-----------------------------------------------------------------------002671
CDECK WPOLE                                                             002672
      SUBROUTINE WPOLE(WREAL,WIMAG,YPLUSR,YPLUSI,YN1R,YN1I,             002673
     * YN2R,YN2I,YP1R,YP1I,YP2R,YP2I)                                   002674
C-----------------------------------------------------------------------002675
C                                                                       002676
C     THIS SUBROUTINE CALCULATES THE W OR W' LOW-RATE POLE              002677
C     FROM THE HIGH-RATE POLE USING THE FOLLOWING:                      002678
C                                                                       002679
C         W  = -A * [(1-Y**XNT)/(1+Y**XNT)]                             002680
C                                                                       002681
C     WHERE                                                             002682
C                                                                       002683
C         Y**XNT = [(AP+WP)/(AP-WP)]**XNT                               002684
C         W  = LOW-RATE W OR W' VARIABLE  [Z=EXP(ST)]                   002685
C         WP = HIGH-RATE W OR W' VARIABLE  [Z=EXP(ST/N)]                002686
C                                                                       002687
C         W TRANSFORM:                                                  002688
C                   A = 1.0                                             002689
C                   AP = 1.0                                            002690
C                   W = W                                               002691
C                   WP = WP                                             002692
C                                                                       002693
C          W' TRANSFORM:                                                002694
C                   A = 2/TEXT                                          002695
```

```
C                    AP = 2/TINT                                                002696
C                    W = W'                                                     002697
C                    WP = WP'                                                   002698
C                                                                               002699
C       TEXT = LOW-RATE SAMPLING PERIOD (SEC)                                   002700
C       TEXT = HIGH-RATE SAMPLING PERIOD (SEC)                                  002701
C       XNT = TEXT/TINT                                                         002702
C                                                                               002703
C       WP = NUMERICAL VALUE OF HIGH-RATE W OR W'                               002704
C            POLE TO BE CONVERTED TO LOW-RATE POLE.                             002705
C                                                                               002706
C    THE PROCEDURE IS TO SIMPLY INSERT THE VALUE OF THE                         002707
C    HIGH-RATE POLE WP = WREAL,WIMAG INTO THE ABOVE                             002708
C    EXPRESSION WITH THE APPROPRIATE VALUES OF A AND AP                         002709
C    TO DETERMINE THE VALUE OF THE LOW-RATE POLE.                               002710
C                                                                               002711
C    ALSO, THE FOLLOWING TERMS  ARE OUTPUTTED FOR USE IN                        002712
C    THE RESIDUE CALCULATIONS.                                                  002713
C                                                                               002714
C          YPLUSR,YPLUSI = (1 + Y**XNT)                                         002715
C          YN1R,YN1I = Y**(XNT-1)                                               002716
C          YN2R,YN2I = Y**(XNT-2)                                               002717
C          YP1R,YP1I = Y'                                                       002718
C          YP2R,YP2I = Y''                                                      002719
C                                                                               002720
C                                                                               002721
      DOUBLE WREAL,WIMAG,YPLUSR,YPLUSI,REAL,REAL1,RIMAG,RIMAG1,REAL2            002722
      DOUBLE YN1R,YN1I,YN2R,YN2I,YP1R,YP1I,YP2R,YP2I,DBCLK,RIMAG2               002723
      DOUBLE DXA,DXAP,ARD,ARN,TINT,TEXT,XNT,YMINR,YMINI,C,ANG                   002724
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,K2ERO,NCLZDB,NCLPDB,DBCLK                 002725
C                                                                               002726
      KXNT=0                                                                    002727
C  SET VALUE OF   'A'=DXA  AND 'AP'=DXAP  FOR W OR W' TRANSFORM                 002728
      DXA=1.0                                                                   002729
      DXAP=1.0                                                                  002730
      IF(TXFORM .EQ. 2HWP) DXA=2.0/TEXT                                         002731
      IF(TXFORM .EQ. 2HWP) DXAP=2.0/TINT                                        002732
C  TRANSFER INTEGER VALUE OF XNT INTO NT                                        002733
      NT=XNT                                                                    002734
C  CHECK FOR INTEGER RATIO XNT                                                  002735
      IF((XNT-NT).EQ.0.0) KXNT=10                                               002736
C  CALCULATE  Y(REAL,RIMAG) = [(AP+WP)/(AP-WP)]                                 002737
      ARD = DXAP + WREAL                                                        002738
      ARN = DXAP - WREAL                                                        002739
C                                                                               002740
C  CHECK FOR POLE AT W=1.0 OR W'=2/T                                            002741
C  THESE POLES TRANSFORM BACK INTO                                              002742
C  THE Z-PLANE AT  Z=INFINITY                                                   002743
C                                                                               002744
      IF(ARN .EQ. 0.0 .AND. WIMAG .EQ. 0.0) GO TO 500                           002745
C                                                                               002746
      CALL DIVI(ARD,WIMAG,ARN,-WIMAG,REAL,RIMAG)                                002747
C                                                                               002748
C  CALCULATE THE FOLLOWING TERMS USED IN THE NUMERATOR                          002749
C  EXPRESSION FOR THE RESIDUES:                                                 002750
```

```
C                                                                       002751
C                 Y = [(AP+WP)/(AP-WP)]                                 002752
C                                                                       002753
C                 Y**(XNT-1) = YN1R,YN1I                                002754
C                 Y**(XNT-2) = YN2R,YN2I                                002755
C                                                                       002756
C                 Y' = YP1R,YP1I                                        002757
C                 Y'' = YP2R,YP2I                                       002758
C  WHERE                                                                002759
C                                                                       002760
C                 Y' = (2.0*AP)/[(AP-WP)**2]                            002761
C                 Y'' = (4.0*AP)/[(AP-WP)**3]                           002762
C                                                                       002763
C                                                                       002764
      CALL MULT(ARN,-WIMAG,ARN,-WIMAG,YP1R,YP1I)                        002765
      YP2R=YP1R                                                         002766
      YP2I=YP1I                                                         002767
      REAL2= 2.0*DXAP                                                   002768
      RIMAG2=0.0                                                        002769
      CALL DIVI(REAL2,RIMAG2,YP2R,YP2I,YP1R,YP1I)                       002770
      CALL MULT(ARN,-WIMAG,YP2R,YP2I,REAL1,RIMAG1)                      002771
      REAL2= 4.0*DXAP                                                   002772
      RIMAG2=0.0                                                        002773
      CALL DIVI(REAL2,RIMAG2,REAL1,RIMAG1,YP2R,YP2I)                    002774
C                                                                       002775
      YN1R=0.0                                                          002776
      YN1I=0.0                                                          002777
      YN2R=0.0                                                          002778
      YN2I=0.0                                                          002779
      IF(REAL.EQ.0.0.AND.RIMAG.EQ.0.0.AND.XNT.EQ.1.0) YN1R=1.0          002780
      IF(REAL.EQ.0.0.AND.RIMAG.EQ.0.0.AND.XNT.EQ.2.0) YN2R=1.0          002781
      IF(REAL .EQ. 0.0 .AND. RIMAG .EQ. 0.0) GO TO 100                  002782
      C=(REAL*REAL+RIMAG*RIMAG)**((XNT-1.0)*0.5)                        002783
      ANG=(DATAN2(RIMAG,REAL))*(XNT-1.0)                                002784
      YN1R=C*DCOS(ANG)                                                  002785
      YN1I=C*DSIN(ANG)                                                  002786
C                                                                       002787
      C=(REAL*REAL+RIMAG*RIMAG)**((XNT-2.0)*0.5)                        002788
      ANG=(DATAN2(RIMAG,REAL))*(XNT-2.0)                                002789
      YN2R=C*DCOS(ANG)                                                  002790
      YN2I=C*DSIN(ANG)                                                  002791
C                                                                       002792
C                                                                       002793
      IF(KXNT .GT. 5) GO TO 200                                         002794
C  POLE TRANFORM SECTION FOR NON-INTEGER VALUES OF XNT                  002795
C                                                                       002796
C  CALCULATE THE  Y**XNT  MAGNITUDE                                     002797
      C=(REAL*REAL+RIMAG*RIMAG)**(XNT*0.5)                              002798
C  CACULATE THE  Y**XNT  PHASE ANGLE                                    002799
      ANG=(DATAN2(RIMAG,REAL))*XNT                                      002800
C  CALCULATE REAL AND IMAGINARY PART OF  Y**XNT=REAL,RIMAG              002801
      REAL=C*DCOS(ANG)                                                  002802
      RIMAG=C*DSIN(ANG)                                                 002803
      GO TO 300                                                         002804
C  POLE TRANSFORM SECTION FOR INTEGER VALUES OF XNT.                    002805
```

```
C  MULTIPLY "Y" TIMES ITSELF XNT TIMES = Y**XNT                          002806
  200 CONTINUE                                                           002807
      IF(XNT .EQ. 1.0) GO TO 300                                         002808
      REAL1=REAL                                                         002809
      RIMAG1=RIMAG                                                       002810
      NTT=NT-1                                                           002811
      DO 400 K=1,NTT                                                     002812
      REAL2=REAL                                                         002813
      REAL=REAL*REAL1-RIMAG*RIMAG1                                       002814
      RIMAG=REAL2*RIMAG1+REAL1*RIMAG                                     002815
  400 CONTINUE                                                           002816
  300 CONTINUE                                                           002817
C  CALCULATE YMINR,YMINI=1-Y**XNT  AND  YPLUSR,YPLUSI=1+Y**XNT          002818
  100 CONTINUE                                                           002819
      YMINR=1-REAL                                                       002820
      YMINI=-RIMAG                                                       002821
      YPLUSR=1+REAL                                                      002822
      YPLUSI=RIMAG                                                       002823
      IF(XNT .EQ. 1.0) RETURN                                            002824
C  CALCULATE THE LOW-RATE POLE  W = -A*[(1-Y**XNT)/(1+Y**XNT)]           002825
      YMINR= -DXA*YMINR                                                  002826
      YMINI= -DXA*YMINI                                                  002827
      CALL DIVI(YMINR,YMINI,YPLUSR,YPLUSI,WREAL,WIMAG)                   002828
      GO TO 50                                                           002829
  500 WRITE(7,1)                                                         002830
    1 FORMAT(1H ,*POLE AT  W=1.0  OR W'=2/T TRANSFORM*,                  002831
     +* BACK INTO Z-PLANE AT  Z=INFINITY*,/,* EXIT PROGRAM*)             002832
      CALL DUMP                                                          002833
   50 RETURN                                                             002834
      END                                                                002835
C                                                                        002836
C                                                                        002837
C------------------------------------------------------------------------ 002838
CDECK ZMULT1                                                             002839
      OVERLAY(27,1)                                                      002840
      PROGRAM ZMULT1                                                     002841
C------------------------------------------------------------------------ 002842
C                                                                        002843
C------------------------------------------------------------------------ 002844
C     SUBROUTINE ZMULT1                                                  002845
C------------------------------------------------------------------------ 002846
C                                                                        002847
C   THIS SUBROUTINE CALCULATES THE RESIDUES FOR POLES WITH               002848
C   MULTIPLICITY EQUAL TO ONE (1).  THE GENERAL EXPRESSION               002849
C   USED IS                                                              002850
C                                                                        002851
C     RESIDUE = [ N(P)*Z ] / [ D1(P)'*Z - (D1(P) * P**XNT)' ]            002852
C           [ P=POLES G(P)/P ]                                           002853
C                                                                        002854
C     RESIDUE = Z* [N(P)/D1(P)'] / [Z-(D1(P) * P**XNT)'/D1(P)']          002855
C           [ P=POLES G(P)/P ]                                           002856
C                                                                        002857
C         G(P) = N(P)/D(P)  ---  HIGH-RATE TXFORM  P=Z=EXP(S*T/N)        002858
C         Z = EXP(S*T)  ---  DEFINITION LOW-RATE TXFORM                  002859
C         D1(P) = D(P) * P                                               002860
```

```
C         PRIME  (   )'   ---   DENOTES FIRST DERIVATIVE WRT 'P'
C         XNT = RATIO HIGH-RATE TO LOW-RATE SAMPLING INTERVALS
C
C    IN THE SUBROUTINE, THE RESIDUE EXPRESSION IS MECHANIZED USING
C         N(P) = DBNPOLY(102)
C         D(P) = DBDPOLY(106)
C         XNT =  XNT
C
C    THE RESULTING RESIDUES ARE STORED IN THE FOLLOWING ARRAYS:
C
C      RESIDUES = Z* SUM OF [ [RES1K(51,2)] / [Z- RES1POL(51,2)] ]
C           [ Z = EXP(S*T) ]
C
C
      DOUBLE DBNPOLY(102),DBDPOLY(106),DBPOLE(51,2),DBZERO(51,2)
      DOUBLE RES1POL(51,2),RES1K(51,2)
      DOUBLE A(106),B(106),C3(318),C6(318)
      DOUBLE R(2),V1(2),V2(2),X,Y,ZF,REAL,RIMAG
      DOUBLE TINT,TEXT,XNT,DBCLK
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK
      COMMON/TXCONV1/DBNPOLY,DBDPOLY,DBPOLE,DBZERO
      COMMON/TXCONV2/A,B,C3,C6
      COMMON/TXCONV3/RES1K,RES1POL
C
C
C      ********** POLE TXFORM TO LOW-RATE DOMAIN **********
C    CALL SUBROUTINE 'POLE' TO CALCULATE NEW LOCATION OF
C    SIMPLE POLE IN LOW-RATE Z-PLANE, Z=EXP(S*T).  THIS
C    NEW POLE LOCATION IS SIMPLY THE ORIGIN . POLE
C    LOCATION IN THE P-PLANE, Z=EXP(S*T/N),  AKENED TO A
C    POWER EQUAL TO THE RATIO OF THE LOW-RA.. TO HIGH-RATE
C    SAMPLING INTERVALS, XNT.
C
C           [ Z = EXP(S*T) ]          [ Z = EXP(S*T/N) ]
C             [ NEW POLE ]      =     [ ORIGINAL POLE ]**XNT
C
C    THESE NEW POLES ARE STORED IN THE ARRAY RES1POL(51,2)
C    IN THE FORMAT: REAL,IMAG.
C
C   TRANSFER POLE ACCORDING TO  'IPOLE'
C   IPOLE FROM MASTER    DO 400 I=1,NM
      REAL=DBPOLE(IPOLE,1)
      RIMAG=DBPOLE(IPOLE,2)
      CALL POLE(REAL,RIMAG)
C   INCREMENT RESIDUE COUNTER FOR MUL = 1
      JRES1=JRES1+1
C   STORE LOW-RATE POLE IN ARRAY RES1POL(51,2)
      RES1POL(JRES1,1)=REAL
      RES1POL(JRES1,2)=RIMAG
C
C
C      ********** CALCULATE NUMERATOR COEFFICIENTS **********
C    CALCULATE THE NUMERATOR COEFFICIENT TERM FOR EACH SIMPLE
C    RESIDUE AND STORE THE REAL AND IMAGINARY PARTS IN
```

```
C    ARRAY RES1K(51,2).                                                  002916
C                                                                        002917
C   STORE POLE TO BE EVALUATED IN  R                                     002918
       R(1)=DBPOLE(IPOLE,1)                                              002919
       R(2)=DBPOLE(IPOLE,2)                                              002920
C   EVALUATE  N(P)  -  NUMERATOR POLY                                    002921
C   DBNPOLY = POLY TO BE EVALUATED AT 'R'                                002922
C   NN = 2 * (DBNPOLY ORDER + 1)                                         002923
C      = OR NUMBER OF OCCUPIED ELEMENTS IN DBNPOLY                       002924
C   R = POLE TO BE EVALUATED (REAL,IMAG)                                 002925
C   V1 = RESULTS OF EVALUATION (REAL,IMAG)                               002926
C   ZF = SCALE FACTOR FOR NUMERICAL ACCURACY                             002927
C   2  = FORMAT OF DBNPOLY  (REAL COEFF, Z-PWR)                          002928
       CALL EVALU3(DBNPOLY,NN,R,V1,ZF,2)                                 002929
       V1(1)=V1(1)*ZF                                                    002930
       V1(2)=V1(2)*ZF                                                    002931
C OBTAIN D(P)'  -  DENOMINATOR DERIVATIVE                                002932
C   DBDPOLY = POLY TO BE DIFFERENTIATED                                  002933
C   ND = 2 * (DBDPOLY ORDER + 1)                                         002934
C      = OR NUMBER OF OCCUPIED ELEMENTS IN DBDPOLY                       002935
C   A  = ARRAY TO CONTAIN THE DERIVATIVE                                 002936
C   NA = 2 * (ORDER OF 'A' + 1)                                          002937
C      = OR NUMBER OF OCCUPIED ELEMENTS IN ARRAY  'A'                    002938
C   2  = FORMAT OF DBDPOLY  (REAL, Z-PWR)                                002939
       CALL DERIV3 (DBDPOLY,ND,A,NA,2)                                   002940
C                                                                        002941
C   EVALUATE D(P)' AT THE POLE 'R'                                       002942
       CALL EVALU3(A,NA,R,V2,ZF,2)                                       002943
       V2(1)=V2(1)*ZF                                                    002944
       V2(2)=V2(2)*ZF                                                    002945
C   CALCULATE EVALUATED EXPRESSION  N(P)/D(P)'                           002946
C   THE REAL,IMAG VALUES ARE SIMPLY DEVIDED                              002947
       CALL DIVI(V1(1),V1(2),V2(1),V2(2),X,Y)                            002948
C                                                                        002949
C   STORE NUMERATOR TERM IN ARRAY  RES1K(51,2)                           002950
       RES1K(JRES1,1)=X                                                  002951
       RES1K(JRES1,2)=Y                                                  002952
       RETURN                                                            002953
       END                                                               002954
C                                                                        002955
C                                                                        002956
C------------------------------------------------------------------------ 002957
CDECK UMULT1                                                             002958
       OVERLAY(27,2)                                                     002959
       PROGRAM UMULT1                                                    002960
C------------------------------------------------------------------------ 002961
C                                                                        002962
C------------------------------------------------------------------------ 002963
C      SUBROUTINE UMULT1                                                 002964
C------------------------------------------------------------------------ 002965
C                                                                        002966
C                                                                        002967
C      THIS ROUTINE CALCULATES THE RESIDUES FOR POLES WITH               002968
C      MULTIPLICITY EQUAL TO ONE (1).  THE EXPRESSION USED               002969
C      IS:                                                               002970
```

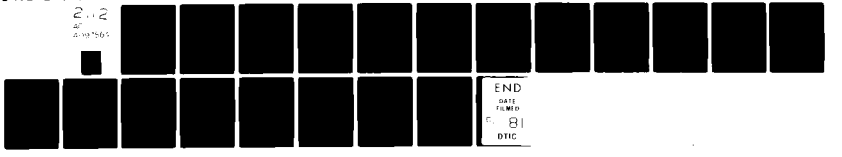
```
C                                                                       002971
C     NUMERATOR = 2AP*[N(UP)/D'(UP)']*[(A+U)/(1+Y**XNT)]                002972
C     DENOMINATOR = U + A*[(1-Y**XNT)/(1+Y**XNT)]                       002973
C                                                                       002974
C     WHERE                                                             002975
C                                                                       002976
C         D'(UP)' = [D(UP)*(AP+UP)*(AP-UP)]'                            002977
C         D'(UP)' = DERIVATIVE WITH RESPECT TO UP                       002978
C                                                                       002979
C         Y = [(AP+UP)/(AP-UP)]                                         002980
C         Y**XNT = [(AP+UP)/(AP-UP)]**XNT                               002981
C                                                                       002982
C         U = LOW-RATE W OR W' VARIABLE  [Z = EXP(ST)]                  002983
C         UP = HIGH-RATE U OR U' VARIABLE  [Z = EXP(ST/N)]              002984
C                                                                       002985
C         BILINEAR TRANSFORMATION BETWEEN Z, U, AND U' PLANES:          002986
C             U = (Z-1)/(Z+1)          U' = (2/T)*(Z-1)/(Z+1)           002987
C             Z = (1+U)/(1-U)          Z = (2/T+U')/(2/T-U')            002988
C                                                                       002989
C         U TRANSFORM:                                                  002990
C                    A = 1.0                                            002991
C                   AP = 1.0                                            002992
C                    U = U                                              002993
C                   UP = UP                                             002994
C                                                                       002995
C         U' TRANSFORM:                                                 002996
C                    A = 2/TEXT                                         002997
C                   AP = 2/TINT                                         002998
C                    U = U'                                             002999
C                   UP = UP'                                            003000
C                                                                       003001
C         TEXT = LOW-RATE SAMPLING PERIOD (SEC)                         003002
C         TINT = HIGH-RATE SAMPLING PERIOD (SEC)                        003003
C         XNT = TEXT/TINT                                               003004
C                                                                       003005
C                                                                       003006
C     IN THE SUBROUTINE, THE NUMERATOR TERM FOR THE                     003007
C     RESIDUE IS MECHANIZED VIA:                                        003008
C                                                                       003009
C         RES1K(51,2) = 2AP*[N(UP)/D'(UP)']*[1+Y**XNT]                  003010
C         DBNPOLY(102) = N(UP)                                          003011
C         DBDPOLY(106)]' = D'(UP)'                                      003012
C         YPLUSR,YPLUSI = [1+Y**XNT]                                    003013
C                                                                       003014
C     CALCULATION OF THE LOW-RATE POLE FROM THE HIGH-RATE               003015
C     POLE IS ACCOMPLISHED IN THE SUBROUTINE  'WPOLE'.                  003016
C     THESE SIMPLE LOW-RATE POLES ARE STORED IN                         003017
C     RES1POL(51,2).  THAT IS,                                          003018
C                                                                       003019
C         RES1POL(51,2) = -A*[(1-Y**XNT)/(1+Y**XNT)]                    003020
C         Y**XNT = [(AP+UP)/(AP-UP)]**XNT                               003021
C         UP = NUMERICAL VALUE OF HIGH-RATE POLE                        003022
C                                                                       003023
C                                                                       003024
      DOUBLE DBNPOLY(102),DBDPOLY(106),DBPOLE(51,2),DBZERO(51,2)        003025
```

```
      DOUBLE RES1POL(51,2),RES1K(51,2)                                  003026
      DOUBLE A(106),B(106),C3(318),C6(318)                              003027
      DOUBLE WREAL,WIMAG,R(2),DXAP,V1(2),V2(2),ZF,X,Y                   003028
      DOUBLE YN1R,YN1I,YN2R,YN2I,YP1R,YP1I,YP2R,YP2I                    003029
      DOUBLE TINT,TEXT,XNT,YPLUSR,YPLUSI,DBCLK                          003030
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER                003031
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK         003032
      COMMON/TXCONV1/DBNPOLY,DBDPOLY,DBPOLE,DBZERO                      003033
      COMMON/TXCONV2/A,B,C3,C6                                          003034
      COMMON/TXCONV3/RES1K,RES1POL                                      003035
C                                                                       003036
C     **********  POLE TRANSFORM TO LOW-RATE  **********               003037
C     CALL SUBROUTINE WPOLE TO CALCULATE LOW-RATE POLE                  003038
C     IN W OR W' PLANE FROM HIGH-RATE POLE IN W OR W'                   003039
C     PLANE.  THESE LOW-RATE POLES ARE STORED IN ARRAY                  003040
C     RES1POL(51,2) IN THE FORMAT: REAL,IMAG.                           003041
C                                                                       003042
C  TRANSFER POLE ACCORDING TO "IPOLE" VARIABLE FROM MASTER              003043
C  DO LOOP  400 I=1,NM IN THE MAIN PROGRAM.                             003044
      WREAL=DBPOLE(IPOLE,1)                                             003045
      WIMAG=DBPOLE(IPOLE,2)                                             003046
      CALL WPOLE(WREAL,WIMAG,YPLUSR,YPLUSI,YN1R,YN1I,                   003047
     1 YN2R,YN2I,YP1R,YP1I,YP2R,YP2I)                                   003048
C  INCREMENT RESIDUE COUNTER FOR  MULTIPLICITY = 1.                     003049
      JRES1=JRES1+1                                                     003050
C  STORE LOW-RATE POLE IN ARRAY RES1POL(51,2)                           003051
      RES1POL(JRES1,1)=WREAL                                            003052
      RES1POL(JRES1,2)=WIMAG                                            003053
C                                                                       003055
C     *****  CALCULATE NUMERATOR CONSTANT IN RESIDUE  *****             003056
C     CALCULATE NUMERATOR CONSTANT FOR EACH RESIDUE AND                 003057
C     STORE THE REAL AND IMAGINARY PARTS IN ARRAY RES1K(51,2).          003058
C          [FOR SUBROUTINES 'EVALU3, 'DERIV3', AND                      003059
C          'DIVI', SEE SUBROUTINE ZMULT1 FOR                            003060
C          EXPLANATION OF SUBROUTINE ARGUMENTS]                         003061
C                                                                       003062
C  STORE POLE TO BE EVALUATED IN ARRAY R(2)                             003063
      R(1)=DBPOLE(IPOLE,1)                                              003064
      R(2)=DBPOLE(IPOLE,2)                                              003065
      DXAP=1.0                                                          003066
      IF(TXFORM .EQ. 2HUP) DXAP=2.0/TINT                                003067
      CALL EVALU3(DBNPOLY,NN,R,V1,ZF,2)                                 003068
      V1(1)=V1(1)*ZF                                                    003069
      V1(2)=V1(2)*ZF                                                    003070
      CALL DERIV3(DBDPOLY,ND,A,NA,2)                                    003071
      CALL EVALU3(A,NA,R,V2,ZF,2)                                       003072
      V2(1)=V2(1)*ZF                                                    003073
      V2(2)=V2(2)*ZF                                                    003074
      CALL DIVI(V1(1),V1(2),V2(1),V2(2),X,Y)                            003075
      X=X*(2.0*DXAP)                                                    003076
      Y=Y*(2.0*DXAP)                                                    003077
      CALL DIVI(X,Y,YPLUSR,YPLUSI,V1(1),V1(2))                          003078
      RES1K(JRES1,1)=V1(1)                                              003079
      RES1K(JRES1,2)=V1(2)                                              003080
```

```
      IF(KZERO .LT. 5) GO TO 100                                        003081
      X=R(1)+DXAP                                                       003082
      Y=R(2)                                                            003083
      CALL DIVI(V1(1),V1(2),X,Y,V2(1),V2(2))                            003084
      RES1K(JRES1,1)=V2(1)                                              003085
      RES1K(JRES1,2)=V2(2)                                              003086
  100 CONTINUE                                                          003087
      RETURN                                                            003088
      END                                                               003089
C                                                                       003090
C                                                                       003091
C-----------------------------------------------------------------------003092
CDECK WMULT2                                                            003093
      OVERLAY(27,3)                                                     003094
      PROGRAM WMULT2                                                    003095
C-----------------------------------------------------------------------003096
C                                                                       003097
C-----------------------------------------------------------------------003098
C     SUBROUTINE WMULT2                                                 003099
C-----------------------------------------------------------------------003100
C                                                                       003101
C                                                                       003102
C     THIS SUBROUTINE CALCULATES THE RESIDUES FOR POLES WITH            003103
C     MULTIPLICITY EQUAL TO TWO (2).  THE EXPRESSION USED IS:           003104
C                                                                       003105
C              [[[[  SUM  OF RESIDUES OF  ]]]]                          003106
C   2*AP*(A+W) *[(NUM1+NUM2+NUM3)*W + (NUM4+NUM5+NUM6)]  -  NUMERATOR   003107
C   [ W + A*[(1-Y**XNT)/(1+Y**XNT)] ]**2  -  DENOMINATOR                003108
C              [[[[  EVALUATED AT WP=HIGH-RATE POLES  ]]]]              003109
C                                                                       003110
C                                                                       003111
C          Y = [(AP+WP)/(AP-WP)]                                        003112
C          XNT = TEXT/TINT                                              003113
C          TEXT = LOW-RATE SAMPLING PERIOD (SEC)                        003114
C          TINT = HIGH-RATE SAMPLING PERIOD (SEC)                       003115
C                                                                       003116
C                                                                       003117
C     NUM1 = [(2*N')/((D*'')*(1+Y**XNT))]                               003118
C     NUM2 = [(-2*N*D*'')/((3)*(D*'')*(D*'')*(1+Y**XNT))]               003119
C     NUM3 = [((-2*N*XNT)*(Y**XNT-1)*(Y'))/((D*'')*(1+Y**XNT)**2))]     003120
C     NUM4 = [((2*N')*(POLE))/((D*'')*(1+Y**XNT))]                      003121
C     NUM5 = [((-2*N*D*'')*(POLE))/((3)*(D*'')*(D*'')*(1+Y**XNT))]      003122
C     NUM6 = [((2*N*A*XNT)*(Y**XNT-1)*(Y'))/((D*'')*(1+Y**XNT)**2))]    003123
C                                                                       003124
C          G(WP) = N(WP)/D(WP)  --  HIGH-RATE TRANSFER FUNCTION         003125
C          G(WP) = N/D                                                  003126
C          D* = D*(WP) = D * (AP-WP) * (AP+WP)                          003127
C          LOW-RATE POLE =  -A*[(1-Y**XNT)/(1+Y**XNT)]                  003128
C          POLE = A*[(1-Y**XNT)/(1+Y**XNT)]                             003129
C                                                                       003130
C                                                                       003131
C          Y**XNT =  [(AP+WP)/(AP-WP)]**XNT                             003132
C          Y**XNT-1 = [(AP+WP)/(AP-WP)]**(XNT-1)                        003133
C          Y' = DERIVATIVE OF Y WITH RESPECT TO 'WP'                    003134
C             = (2.0*AP)/[(AP-WP)**2]                                   003135
```

```
C                                                                       003136
C          N = NUMERATOR POLYNOMIAL (HIGH-RATE)                         003137
C          N' = FIRST DERIVATIVE OF N WITH RESPECT TO 'WP'              003138
C          D = DENIMINATOR POLYNOMIAL (HIGH-RATE)                       003139
C          D# = D * (AP-LP) * (AP+WP)                                   003140
C          D#'' = SECOND DERIVATIVE                                     003141
C          D#''' = THIRD DERIVATIVE                                     003142
C                                                                       003143
C          W TRANSFORM:                                                 003144
C                    A=1.0                                              003145
C                    AP=1.0                                             003146
C                    WP=WP = HIGH-RATE TRANSFROM VARIABLE               003147
C                    W=W = LOW-RATE TRANSFROM VARIABLE                  003148
C                                                                       003149
C          W' TRANSFORM:                                                003150
C                    A=2/TEXT                                           003151
C                    AP=2/TINT                                          003152
C                    WP=WP' = HIGH-RATE TRANSFROM VARIABLE              003153
C                    W=W' = LOW-RATE TRANSFROM VARIABLE                 003154
C                                                                       003155
C                                                                       003156
      DOUBLE DBNPOLY(102),DBDPOLY(106),DBPOLE(51,2),DBZERO(51,2)        003157
      DOUBLE A(106),B(106),C3(318),C6(318)                              003158
      DOUBLE R(2),DXAP,DXA,V1(2),V2(2),V3(2),ZF,X,Y                     003159
      DOUBLE VW1(2),VW2(2),VW3(2)                                       003160
      DOUBLE VNUM(2),VNUM1(2),VDEN2(2),VDEN3(2)                         003161
      DOUBLE WREAL,WIMAG,YPLUSR,YPLUSI,YN1R,YN1I,YN2R,YN2I,             003162
     1 YP1R,YP1I,YP2R,YP2I                                              003163
      DOUBLE TINT,TEXT,XNT,DBCLK                                        003164
      DOUBLE RES2K1(25,2),RES2K2(25,2),RES2POL(25,2)                    003165
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER                003166
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK         003167
      COMMON/TXCONV1/DBNPOLY,DBDPOLY,DBPOLE,DBZERO                      003168
      COMMON/TXCONV2/A,B,C3,C6                                          003169
      COMMON/TXCONV4/RES2K1,RES2K2,RES2POL                              003170
C                                                                       003171
C     ************** POLE TRANSFORM TO LOW-RATE ********************    003172
C     CALL SUBROUTINE WPOLE TO CALCULATE LOW-RATE POLE IN THE           003173
C     W OR W' PLANE.  THESE LOW-RATE POLES ARE STORED IN ARRAY          003174
C     RES2POLE(25,2) IN THE FORMAT: REAL,IMAG                           003175
C                                                                       003176
C   TRANSFER POLE ACCORDING TO THE 'IPOLE' VARIABLE FROM MASTER         003177
C   DO LOOP  400 I=1,NM  IN THE MAIN PROGRAM.                           003178
      WREAL=DBPOLE(IPOLE,1)                                             003179
      WIMAG=DBPOLE(IPOLE,2)                                             003180
      CALL WPOLE(WREAL,WIMAG,YPLUSR,YPLUSI,YN1R,YN1I,                   003181
     1 YN2R,YN2I,YP1R,YP1I,YP2R,YP2I)                                   003182
C                                                                       003183
C   INCREMENT RESIDUE COUNTER FOR MULTIPLICITY = 2                      003184
      JRES2=JRES2+1                                                     003185
C   STORE LOW-RATE POLE IN ARRAY RES2POL(25,2)                          003186
      RES2POL(JRES2,1)=WREAL                                            003187
      RES2POL(JRES2,2)=WIMAG                                            003188
C                                                                       003189
C     ******* CALCULATE NUMERATOR TERMS IN RESIDUE *******             003190
```

AD-A097 864 SYSTEMS TECHNOLOGY INC HAWTHORNE CA F/G 9/2
MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS--ETC(U)
SEP 80 D G DIDALEUSKY F33615-79-C-3601
UNCLASSIFIED STI-TR-1142-1-3 AFWAL-TR-80-3101-VOL-3 NL



END
DATE FILMED
81
DTIC

```
C     CALCULATE THE INDIVIUAL NUMERATOR TERMS IN THE                    003191
C     RESIDUE AND STORE THEM AS:                                        003192
C                                                                       003193
C     [RES2K1(25,2)]*U + [RES2K2(25,2)] / U - [RES2POL(25,2)]           003194
C                                                                       003195
C  STORE HIGH-RATE POLE TO BE EVALUATED IN ARRAY R(2)                   003196
      R(1)=DBPOLE(IPOLE,1)                                              003197
      R(2)=DBPOLE(IPOLE,2)                                              003198
      DXA=1.0                                                           003199
      DXAP=1.0                                                          003200
      IF(TXFORM .EQ. 2HUP) DXA=2.0/TEXT                                 003201
      IF(TXFORM .EQ. 2HUP) DXAP=2.0/TINT                                003202
C                                                                       003203
C  OBTAIN:   N(UP)=VNUM(2);  N(UP)''=VNUM1(2);  D*(UP)''=VDEN2(2);      003204
C            AND  D*(UP)'''=VDEN3(2)                                    003205
      CALL EVALU3(DBNPOLY,NN,R,VNUM,ZF,2)                               003206
      VNUM(1)=VNUM(1)*ZF                                                003207
      VNUM(2)=VNUM(2)*ZF                                                003208
      CALL DERIV3(DBNPOLY,NN,A,NA,2)                                    003209
      CALL EVALU3(A,NA,R,VNUM1,ZF,2)                                    003210
      VNUM1(1)=VNUM1(1)*ZF                                              003211
      VNUM1(2)=VNUM1(2)*ZF                                              003212
      CALL DERIV3(DBDPOLY,ND,A,NA,2)                                    003213
      CALL DERIV3(A,NA,B,NB,2)                                          003214
      CALL EVALU3(B,NB,R,VDEN2,ZF,2)                                    003215
      VDEN2(1)=VDEN2(1)*ZF                                              003216
      VDEN2(2)=VDEN2(2)*ZF                                              003217
      CALL DERIV3(B,NB,A,NA,2)                                          003218
      CALL EVALU3(A,NA,R,VDEN3,ZF,2)                                    003219
      VDEN3(1)=VDEN3(1)*ZF                                              003220
      VDEN3(2)=VDEN3(2)*ZF                                              003221
C                                                                       003222
C  ADD (AP+UP) CANCELLING FACTOR TO NUMERATOR IF                        003223
C  KZERO .GT. 0.  THE VALUE OF KZERO IS SET IN                          003224
C  THE (AP+UP) SECTION OF THE MAIN PROGRAM.                             003225
C  THIS CANCELLATION OCCURS WHEN THE NUMERATOR                          003226
C  CONTAINS THE (AP+UP) ROOT.                                           003227
C                                                                       003228
C     N = N/(AP+UP)                                                     003229
C                                                                       003230
C     N' = [(AP+UP)N' - N]/[(AP+UP)**2]                                 003231
C        = [N'/(AP+UP)] - [N/(AP+UP)**2]                                003232
C                                                                       003233
      IF(KZERO .LT. 5) GO TO 100                                        003234
      V1(1) = DXAP+R(1)                                                 003235
      V1(2) = R(2)                                                      003236
      V2(1) = VNUM(1)                                                   003237
      V2(2) = VNUM(2)                                                   003238
      CALL DIVI(V2(1),V2(2),V1(1),V1(2),VNUM(1),VNUM(2))                003239
      CALL DIVI(VNUM1(1),VNUM1(2),V1(1),V1(2),X,Y)                      003240
      CALL DIVI(VNUM(1),VNUM(2),V1(1),V1(2),V3(1),V3(2))                003241
      VNUM1(1)=X-V3(1)                                                  003242
      VNUM1(2)=Y-V3(2)                                                  003243
  100 CONTINUE                                                          003244
C                                                                       003245
```

```
C  OBTAIN:   [RES2K1(25,2)]*U                                          003246
C                                                                     003247
      V1(1)=2.0*VNUM1(1)                                              003248
      V1(2)=2.0*VNUM1(2)                                              003249
      CALL MULT(VDEN2(1),VDEN2(2),YPLUSR,YPLUSI,V2(1),V2(2))          003250
      CALL DIVI(V1(1),V1(2),V2(1),V2(2),VW1(1),VW1(2))                003251
C                                                                     003252
      V1(1)=-2.0*VNUM(1)                                              003253
      V1(2)=-2.0*VNUM(2)                                              003254
      CALL MULT(V1(1),V1(2),VDEN3(1),VDEN3(2),V2(1),V2(2))            003255
      V1(1)=3.0*YPLUSR                                                003256
      V1(2)=3.0*YPLUSI                                                003257
      CALL MULT(V1(1),V1(2),VDEN2(1),VDEN2(2),V3(1),V3(2))            003258
      CALL MULT(V3(1),V3(2),VDEN2(1),VDEN2(2),V1(1),V1(2))            003259
      CALL DIVI(V2(1),V2(2),V1(1),V1(2),VW2(1),VW2(2))                003260
C                                                                     003261
      V1(1)=-2.0*VNUM(1)                                              003262
      V1(2)=-2.0*VNUM(2)                                              003263
      CALL MULT(VDEN2(1),VDEN2(2),YPLUSR,YPLUSI,V3(1),V3(2))          003264
      CALL MULT(V3(1),V3(2),YPLUSR,YPLUSI,V2(1),V2(2))                003265
      CALL DIVI(V1(1),V1(2),V2(1),V2(2),V3(1),V3(2))                  003266
C                                                                     003267
      V1(1)=XNT*YN1R                                                  003268
      V1(2)=XNT*YN1I                                                  003269
      CALL MULT(V1(1),V1(2),YP1R,YP1I,V2(1),V2(2))                    003270
      CALL MULT(V3(1),V3(2),V2(1),V2(2),VW3(1),VW3(2))                003271
C                                                                     003272
      RES2K1(JRES2,1)=2.0*DXAP*(VW1(1)+VW2(1)+VW3(1))                 003273
      RES2K1(JRES2,2)=2.0*DXAP*(VW1(2)+VW2(2)+VW3(2))                 003274
C                                                                     003275
C  OBTAIN:    [RES2K2(25,2)]                                          003276
C                                                                     003277
      V1(1)=2.0*VNUM1(1)                                              003278
      V1(2)=2.0*VNUM1(2)                                              003279
      V2(1)=-RES2POL(JRES2,1)                                         003280
      V2(2)=-RES2POL(JRES2,2)                                         003281
      CALL MULT(V1(1),V1(2),V2(1),V2(2),V3(1),V3(2))                  003282
      CALL MULT(VDEN2(1),VDEN2(2),YPLUSR,YPLUSI,V1(1),V1(2))          003283
      CALL DIVI(V3(1),V3(2),V1(1),V1(2),VW1(1),VW1(2))                003284
C                                                                     003285
      V1(1)=-2.0*VNUM(1)                                              003286
      V1(2)=-2.0*VNUM(2)                                              003287
      CALL MULT(V1(1),V1(2),VDEN3(1),VDEN3(2),V3(1),V3(2))            003288
      V2(1)=-RES2POL(JRES2,1)                                         003289
      V2(2)=-RES2POL(JRES2,2)                                         003290
      CALL MULT(V3(1),V3(2),V2(1),V2(2),V1(1),V1(2))                  003291
      V2(1)=3.0*YPLUSR                                                003292
      V2(2)=3.0*YPLUSI                                                003293
      CALL MULT(V2(1),V2(2),VDEN2(1),VDEN2(2),V3(1),V3(2))            003294
      CALL MULT(V3(1),V3(2),VDEN2(1),VDEN2(2),V2(1),V2(2))            003295
      CALL DIVI(V1(1),V1(2),V2(1),V2(2),VW2(1),VW2(2))                003296
C                                                                     003297
      V1(1)=2.0*DXA*XNT*VNUM(1)                                       003298
      V1(2)=2.0*DXA*XNT*VNUM(2)                                       003299
      CALL MULT(V1(1),V1(2),YN1R,YN1I,V2(1),V2(2))                    003300
```

```
      CALL MULT(U2(1),U2(2),YP1R,YP1I,U1(1),U1(2))                      003301
      CALL MULT(UDEN2(1),UDEN2(2),YPLUSR,YPLUSI,U2(1),U2(2))            003302
      CALL MULT(U2(1),U2(2),YPLUSR,YPLUSI,U3(1),U3(2))                  003303
      CALL DIVI(U1(1),U1(2),U3(1),U3(2),UU3(1),UU3(2))                  003304
C                                                                       003305
      RES2K2(JRES2,1)=2.0*DXAP*(UU1(1)+UU2(1)+UU3(1))                   003306
      RES2K2(JRES2,2)=2.0*DXAP*(UU1(2)+UU2(2)+UU3(2))                   003307
C                                                                       003308
      RETURN                                                            003309
      END                                                               003310
C                                                                       003311
C                                                                       003312
C-----------------------------------------------------------------------003313
C                                                                       003314
CDECK UMULT3                                                            003315
      OVERLAY(27,4)                                                     003316
      PROGRAM UMULT3                                                    003317
C-----------------------------------------------------------------------003318
C                                                                       003319
C-----------------------------------------------------------------------003320
C     SUBROUTINE UMULT3                                                 003321
C-----------------------------------------------------------------------003322
C                                                                       003323
C                                                                       003324
C     THIS SUBROUTINE CALCULATES THE RESIDUES FOR POLES WITH            003325
C     MULTIPLICITY EQUAL TO THREE(3).  THE EXPRESSION USED IS:          003326
C                                                                       003327
C               CCC SUM OF RESIDUES OF 333                              003328
C                                                                       003329
C    2*AP*(A+U)*C  (UN1+UN2+UN3+UN4)*(U**2+(2*P)*U+P*P)                 003330
C                + (UN5+UN6+UN7+UN8+UN9)*(U**2+(P-A)*U-A*P)             003331
C                + (UN10)*(U**2-(2*A)*U+A*A) ] / C(U+P)**3]             003332
C                                                                       003333
C          P = -(LOW-RATE POLES)                                        003334
C            = A*C(1-Y**XNT)/(1+Y**XNT)]                                003335
C                                                                       003336
C          UN1 - UN10 = NUMERATOR TERMS DEFINED INDIVIDUALLY          003337
C                       IN SUBROUTINE.                                003338
C                                                                       003339
C          Y = C(AP+UP)/(AP-UP)]                                        003340
C          (1+Y**XNT) =  YPLUSR,YPLUSI                                  003341
C          Y**(XNT-1) =  YN1R,YN1I                                      003342
C          Y**(XNT-2) =  YN2R,YN2I                                      003343
C          Y' = YP1R,YP1I                                               003344
C          Y'' = YP2R,YP2I                                              003345
C                                                                       003346
C          XNT=TEXT/TINT                                                003347
C          TEXT = LOW-RATE SAMPLING PERIOD (SEC)                        003348
C          TINT = HIGH-RATE SAMPLING PERIOD (SEC)                       003349
C                                                                       003350
C          G(UP) = N/D                                                  003351
C                = HIGH-RATE TRANSFER FUNCTION                          003352
C          D* = D * (AP+UP) * (AP-UP)                                   003353
C          N = NUMERATOR POLY (HIGH-RATE)                               003354
C          D***' = THIRD DERIVATIVE OF DENOMINATOR POLY WITH            003355
C                  RESPECT TO UP VARIABLE.
```

```
C          D''''' = FOURTH DRIVATIVE.                                          003356
C          D''''' = FIFTH DERIVATIVE.                                          003357
C          N' = FIRST DERIVATIVE NUMERATOR POLY WITH RESPECT                   003358
C               TO WP VARIABLE.                                                003359
C          N'' = SECOND DERIVATIVE.                                            003360
C                                                                              003361
C          W TRANSFORM:                                                        003362
C                    A=1.0                                                     003363
C                    AP=1.0                                                    003364
C                    WP=WP= HIGH-RATE TRANSFORM VARIABLE                       003365
C                    W=W= LOW-RATE TRANSFORM VARIABLE                          003366
C                                                                              003367
C          W' TRANSFORM:                                                       003368
C                    A=2/TEXT                                                  003369
C                    AP=2/TINT                                                 003370
C                    WP=WP'                                                    003371
C                    W=W'                                                      003372
C                                                                              003373
C                                                                              003374
      DOUBLE DBNPOLY(102),DBDPOLY(106),DBPOLE(51,2),DBZERO(51,2)               003375
      DOUBLE A(106),B(106),C3(318),C6(318)                                     003376
      DOUBLE RES3K1(17,2),RES3K2(17,2),RES3K3(17,2),RES3POL(17,2)              003377
      DOUBLE R(2),DXAP,DXA,V1(2),V2(2),V3(2),V4(2),V5(2),ZF                    003378
      DOUBLE VNUM(2),VNUM1(2),VNUM2(2),VDEN3(2),VDEN4(2),VDEN5(2)              003379
      DOUBLE VD3Y(2),VD35Y(2),VD34Y(2),VD34Y1(2)                               003380
      DOUBLE VN1(2),VN2(2),VN3(2),VN4(2),VN5(2),VN6(2),VN7(2)                  003381
      DOUBLE VN8(2),VN9(2),VN10(2),TINT,TEXT,XNT,DBCLK                         003382
      DOUBLE WREAL,WIMAG,YPLUSR,YPLUSI,YN1R,YN1I,YN2R,YN2I,                    003383
     X YP1R,YP1I,YP2R,YP2I                                                     003384
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER                       003385
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK                003386
      COMMON/TXCONV1/DBNPOLY,DBDPOLY,DBPOLE,DBZERO                             003387
      COMMON/TXCONV2/A,B,C3,C6                                                 003388
      COMMON/TXCONV5/RES3K1,RES3K2,RES3K3,RES3POL                              003389
C                                                                              003390
C                                                                              003391
C     ****************  POLE TRANSFER TO LOW-RATE ********************         003392
C     CALL SUBROUTINE WPOLE TO CALCULATE LOW-RATE POLE IN THE                  003393
C     W OR W' PLANE.  THESE LOW-RATE POLES ARE STORED IN ARRAY                 003394
C     RES3POL(17,2) IN THE FORMAT: REAL,IMAG.                                  003395
C                                                                              003396
C     TRANSFER POLE ACCORDING TO THE 'IPOLE' VARIABLE SENT FROM                003397
C     MASTER DO LOOP 400 I=1,NM  IN THE MAIN PROGRAM.                          003398
      WREAL=DBPOLE(IPOLE,1)                                                    003399
      WIMAG=DBPOLE(IPOLE,2)                                                    003400
      CALL WPOLE(WREAL,WIMAG,YPLUSR,YPLUSI,YN1R,YN1I,                          003401
     1 YN2R,YN2I,YP1R,YP1I,YP2R,YP2I)                                          003402
C                                                                              003403
C  INCREMENT RESIDUE COUNTER FOR MULTIPLICITY = 3.                             003404
      JRES3=JRES3+1                                                            003405
C  STORE LOW-RATE POLE IN ARRAY RES3POL(17,2)                                  003406
      RES3POL(JRES3,1)=WREAL                                                   003407
      RES3POL(JRES3,2)=WIMAG                                                   003408
C                                                                              003409
C                                                                              003410
```

```
C     ********  CALCULATE NUMERATOR TERMS IN RESIDUES  ********          003411
C     CALCULATE THE INDIVIDUAL NUMERATOR TERMS IN THE RESIDUES          003412
C     AND STORE THEM AS:                                                003413
C                                                                       003414
C     [RES3K1(17,2)]*(U**2) + [RES3K2(17,2)]*(U) + [RES3K3(17,2)]       003415
C                                                                       003416
C                                                                       003417
C  STORE HIGH-RATE POLE TO BE EVALUATED IN ARRAY R(2)                   003418
      R(1)=DBPOLE(IPOLE,1)                                              003419
      R(2)=DBPOLE(IPOLE,2)                                              003420
      DXA=1.0                                                           003421
      DXAP=1.0                                                          003422
      IF(TXFORM .EQ. 2HUP) DXA=2.0/TEXT                                 003423
      IF(TXFORM .EQ. 2HUP) DXAP=2.0/TINT                                003424
C                                                                       003425
C  OBTAIN  N=VNUM(2);  N'=VNUM1(2);  N''=VNUM2(2);  D***=VDEN3(2)       003426
C          D****=VDEN4(2);  AND  D*****=VDEN5(2)                        003427
C                                                                       003428
      CALL EVALU3(DBNPOLY,NN,R,VNUM,ZF,2)                               003429
      VNUM(1)=VNUM(1)*ZF                                                003430
      VNUM(2)=VNUM(2)*ZF                                                003431
      CALL DERIV3(DBNPOLY,NN,A,NA,2)                                    003432
      CALL EVALU3(A,NA,R,VNUM1,ZF,2)                                    003433
      VNUM1(1)=VNUM1(1)*ZF                                              003434
      VNUM1(2)=VNUM1(2)*ZF                                              003435
      CALL DERIV3(A,NA,B,NB,2)                                          003436
      CALL EVALU3(B,NB,R,VNUM2,ZF,2)                                    003437
      VNUM2(1)=VNUM2(1)*ZF                                              003438
      VNUM2(2)=VNUM2(2)*ZF                                              003439
C                                                                       003440
      CALL DERIV3(DBDPOLY,ND,A,NA,2)                                    003441
      CALL DERIV3(A,NA,B,NB,2)                                          003442
      CALL DERIV3(B,NB,A,NA,2)                                          003443
      CALL EVALU3(A,NA,R,VDEN3,ZF,2)                                    003444
      VDEN3(1)=VDEN3(1)*ZF                                              003445
      VDEN3(2)=VDEN3(2)*ZF                                              003446
      CALL DERIV3(A,NA,B,NB,2)                                          003447
      CALL EVALU3(B,NB,R,VDEN4,ZF,2)                                    003448
      VDEN4(1)=VDEN4(1)*ZF                                              003449
      VDEN4(2)=VDEN4(2)*ZF                                              003450
      CALL DERIV3(B,NB,A,NA,2)                                          003451
      CALL EVALU3(A,NA,R,VDEN5,ZF,2)                                    003452
      VDEN5(1)=VDEN5(1)*ZF                                              003453
      VDEN5(2)=VDEN5(2)*ZF                                              003454
C                                                                       003455
C                                                                       003456
C  ADD (AP+UP) CANCELLING FACTOR TO NUMERATOR IF                        003457
C  KZERO .GT. 0.  THE VALUE OF KZERO IS SET IN                          003458
C  THE (AP+UP) SECTION OF THE MAIN PROGRAM.                             003459
C  THIS CANCELLATION OCCURS WHEN THE NUMERATOR                          003460
C  CONTAINS THE (AP+UP) ROOT.  THE NUMERATOR                            003461
C  TERMS THEN BECOME:                                                   003462
C                                                                       003463
C     N = N/(AP+UP)                                                     003464
C     N' = [N/(AP+UP)]' + [N'/(AP+UP)] - [N/(AP+UP)**2]                 003465
```

```
C      N'' * [N/(AP+UP)]'' * [N'''/(AP+UP)] - [2N''/(AP+UP)**2]                  003466
C                           + [2*N/(AP+UP)**3]                                   003467
C                                                                                003468
       IF(KZERO .LT. 5) GO TO 100                                                003469
       V1(1)=DXAP+R(1)                                                           003470
       V1(2)=R(2)                                                                003471
       V2(1)=VNUM(1)                                                             003472
       V2(2)=VNUM(2)                                                             003473
C                                                                                003474
       CALL DIVI(V2(1),V2(2),V1(1),V1(2),VNUM(1),VNUM(2))                        003475
       V2(1)=VNUM1(1)                                                            003476
       V2(2)=VNUM1(2)                                                            003477
       CALL DIVI(V2(1),V2(2),V1(1),V1(2),VNUM1(1),VNUM1(2))                      003478
       V2(1)=VNUM2(1)                                                            003479
       V2(2)=VNUM2(2)                                                            003480
       CALL DIVI(V2(1),V2(2),V1(1),V1(2),VNUM2(1),VNUM2(2))                      003481
C                                                                                003482
       CALL DIVI(VNUM(1),VNUM(2),V1(1),V1(2),V3(1),V3(2))                        003483
       CALL DIVI(V3(1),V3(2),V1(1),V1(2),V4(1),V4(2))                            003484
       CALL DIVI(VNUM1(1),VNUM1(2),V1(1),V1(2),V5(1),V5(2))                      003485
       VNUM2(1)=VNUM2(1)-2.0*V5(1)+2.0*V4(1)                                     003486
       VNUM2(2)=VNUM2(2)-2.0*V5(2)+2.0*V4(2)                                     003487
       VNUM1(1)=VNUM1(1)-V3(1)                                                   003488
       VNUM1(2)=VNUM1(2)-V3(2)                                                   003489
C                                                                                003490
  100 CONTINUE                                                                   003491
C                                                                                003492
C  OBTAIN COMMON FACTORS IN THE NUMERATOR TERMS.                                 003493
C                                                                                003494
C    VD3Y=[(1+Y**XNT)**-1]*(D*''')**-1                                           003495
C    VD35Y=[(1+Y**XNT)**-1]*[(D*''')**-2]*(D*'''')                               003496
C    VD34Y=[(1+Y**XNT)**-1]*[(D*''')**-2]*(D*'''')                               003497
C    VD34Y1=[(1+Y**XNT)**-1]*[(D*''')**-3]*(D*'''')**2                           003498
C                                                                                003499
       V1(1)=1.0                                                                 003500
       V1(2)=0.0                                                                 003501
       CALL DIVI(V1(1),V1(2),YPLUSR,YPLUSI,V2(1),V2(2))                          003502
       CALL DIVI(V2(1),V2(2),VDEN3(1),VDEN3(2),VD3Y(1),VD3Y(2))                  003503
       CALL DIVI(VD3Y(1),VD3Y(2),VDEN3(1),VDEN3(2),V1(1),V1(2))                  003504
       CALL MULT(V1(1),V1(2),VDEN5(1),VDEN5(2),VD35Y(1),VD35Y(2))                003505
       CALL MULT(V1(1),V1(2),VDEN4(1),VDEN4(2),VD34Y(1),VD34Y(2))                003506
       CALL MULT(VD34Y(1),VD34Y(2),VDEN4(1),VDEN4(2),V1(1),V1(2))                003507
       CALL DIVI(V1(1),V1(2),VDEN3(1),VDEN3(2),VD34Y1(1),VD34Y1(2))              003508
C                                                                                003509
C                                                                                003510
C  OBTAIN THE  'VN1' NUMERATOR TERM                                              003511
C  VN1=3.0*N''*[(1+Y**XNT)**-1]*(D*''')**-1                                      003512
C                                                                                003513
       V1(1)=3.0*VNUM2(1)                                                        003514
       V1(2)=3.0*VNUM2(2)                                                        003515
       CALL MULT(V1(1),V1(2),VD3Y(1),VD3Y(2),VN1(1),VN1(2))                      003516
C                                                                                003517
C  OBTAIN THE  'VN2' NUMERATOR TERM                                              003518
C  VN2=-1.5*N'*[(1+Y**XNT)**-1]*[(D*''')**-2]*D*''''                             003519
C                                                                                003520
```

```
      V1(1)=-1.5*VNUM1(1)                                                    003521
      V1(2)=-1.5*VNUM1(2)                                                    003522
      CALL MULT(V1(1),V1(2),VD34Y(1),VD34Y(2),VN2(1),VN2(2))                 003523
C                                                                            003524
C  OBTAIN THE  'VN3'  NUMERATOR TERM                                         003525
C  VN3=.375*N*[(1+Y**XNT)**-1]*[(D*'')**-3]*[(D*'''')**2]                    003526
C                                                                            003527
      V1(1)=.375*VNUM(1)                                                     003528
      V1(2)=.375*VNUM(2)                                                     003529
      CALL MULT(V1(1),V1(2),VD34Y1(1),VD34Y1(2),VN3(1),VN3(2))               003530
C                                                                            003531
C  OBTAIN THE  'VN4'  NUMERATOR TERM                                         003532
C  VN4=-0.3*N*[(1+Y**XNT)**-1]*[(D*'')**-2]*(D*'''')                         003533
C                                                                            003534
      V1(1)=-0.3*VNUM(1)                                                     003535
      V1(2)=-0.3*VNUM(2)                                                     003536
      CALL MULT(V1(1),V1(2),VD35Y(1),VD35Y(2),VN4(1),VN4(2))                 003537
C                                                                            003538
C  OBTAIN THE  'VN5'  NUMERATOR TERM                                         003539
C  VN5=-6.0*N'*[(1+Y**XNT)**-2]*[(D*'')**-1]*XNT*(Y**XNT-1)*Y'               003540
C                                                                            003541
      V1(1)=-6.0*VNUM1(1)*XNT                                                003542
      V1(2)=-6.0*VNUM1(2)*XNT                                                003543
      CALL MULT(V1(1),V1(2),YN1R,YN1I,V2(1),V2(2))                           003544
      CALL MULT(V2(1),V2(2),YP1R,YP1I,V1(1),V1(2))                           003545
      CALL MULT(V1(1),V1(2),VD3Y(1),VD3Y(2),V2(1),V2(2))                     003546
      CALL DIVI(V2(1),V2(2),YPLUSR,YPLUSI,VN5(1),VN5(2))                     003547
C                                                                            003548
C  OBTAIN THE  'VN6'  NUMERATOR TERM                                         003549
C  VN6=3.0*N*[(1+Y**XNT)**-2]*[(D*'')**-2]*(D*'''')                          003550
C       *(XNT)*[Y**(XNT-1)]*Y'                                               003551
C                                                                            003552
      V1(1)=3.0*VNUM(1)*XNT                                                  003553
      V1(2)=3.0*VNUM(2)*XNT                                                  003554
      CALL MULT(V1(1),V1(2),YN1R,YN1I,V2(1),V2(2))                           003555
      CALL MULT(V2(1),V2(2),YP1R,YP1I,V1(1),V1(2))                           003556
      CALL MULT(V1(1),V1(2),VD34Y(1),VD34Y(2),V2(1),V2(2))                   003557
      CALL DIVI(V2(1),V2(2),YPLUSR,YPLUSI,VN6(1),VN6(2))                     003558
C                                                                            003559
C  OBTAIN THE  'VN7'  NUMERATOR TERM                                         003560
C  VN7=-1.5*N*[(1+Y**XNT)**-2]*[(D*'')**-2]*(D*'''')                         003561
C       *(XNT)*[Y**(XNT-1)]*Y'                                               003562
C                                                                            003563
      V1(1)=-1.5*VNUM(1)*XNT                                                 003564
      V1(2)=-1.5*VNUM(2)*XNT                                                 003565
      CALL MULT(V1(1),V1(2),YN1R,YN1I,V2(1),V2(2))                           003566
      CALL MULT(V2(1),V2(2),YP1R,YP1I,V1(1),V1(2))                           003567
      CALL MULT(V1(1),V1(2),VD34Y(1),VD34Y(2),V2(1),V2(2))                   003568
      CALL DIVI(V2(1),V2(2),YPLUSR,YPLUSI,VN7(1),VN7(2))                     003569
C                                                                            003570
C  OBTAIN THE  'VN8'  NUMERATOR TERM                                         003571
C  VN8=-3.0*N*[(1+Y**XNT)**-2]*[(D*'')**-1]*(XNT)*(XNT-1)                    003572
C     *[Y**(XNT-2)]*[(Y')**2]                                                0C3573
C                                                                            003574
      V1(1)=-3.0*VNUM(1)*XNT*(XNT-1.0)                                       003575
```

```
      V1(2)=-3.0*VNUM(2)*XNT*(XNT-1.0)                                  003576
      CALL MULT(V1(1),V1(2),VN2R,VN2I,V2(1),V2(2))                      003577
      CALL MULT(V2(1),V2(2),VP1R,VP1I,V1(1),V1(2))                      003578
      CALL MULT(V1(1),V1(2),VP1R,VP1I,V2(1),V2(2))                      003579
      CALL MULT(V2(1),V2(2),VD3Y(1),VD3Y(2),V1(1),V1(2))                003580
      CALL DIVI(V1(1),V1(2),VPLUSR,VPLUSI,VN8(1),VN8(2))                003581
C                                                                       003582
C  OBTAIN THE  "VN9"  NUMERATOR TERM                                    003583
C  VN9=-3.0*N*[(1+Y**XNT)**-2]*[(D*'')**-1]*(XNT)*[Y**(XNT-1)]*Y''      003584
C                                                                       003585
      V1(1)=-3.0*VNUM(1)*XNT                                            003586
      V1(2)=-3.0*VNUM(2)*XNT                                            003587
      CALL MULT(V1(1),V1(2),VN1R,VN1I,V2(1),V2(2))                      003588
      CALL MULT(V2(1),V2(2),VP2R,VP2I,V1(1),V1(2))                      003589
      CALL MULT(V1(1),V1(2),VD3Y(1),VD3Y(2),V2(1),V2(2))                003590
      CALL DIVI(V2(1),V2(2),VPLUSR,VPLUSI,VN9(1),VN9(2))                003591
C                                                                       003592
C  OBTAIN THE  "VN10"  NUMERATOR TERM                                   003593
C  VN10=6.0*N*[(1+Y**XNT)**-3]*[(D*'')**-1]*(XNT**2)                    003594
C       *[(Y**(XNT-1))**2]*[(Y')**2]                                    003595
C                                                                       003596
      V1(1)=6.0*VNUM(1)*XNT*XNT                                         003597
      V1(2)=6.0*VNUM(2)*XNT*XNT                                         003598
      CALL MULT(V1(1),V1(2),VN1R,VN1I,V2(1),V2(2))                      003599
      CALL MULT(V2(1),V2(2),VN1R,VN1I,V1(1),V1(2))                      003600
      CALL MULT(V1(1),V1(2),VP1R,VP1I,V2(1),V2(2))                      003601
      CALL MULT(V2(1),V2(2),VP1R,VP1I,V1(1),V1(2))                      003602
      CALL MULT(V1(1),V1(2),VD3Y(1),VD3Y(2),V2(1),V2(2))                003603
      CALL DIVI(V2(1),V2(2),VPLUSR,VPLUSI,V1(1),V1(2))                  003604
      CALL DIVI(V1(1),V1(2),VPLUSR,VPLUSI,VN10(1),VN10(2))              003605
C                                                                       003606
C  COMBINE THE NUMERATOR TERMS TO OBTAIN THE COEFFICIENTS               003607
C  OF THE NUMERATOR OF THE RESIDUE                                      003608
C                                                                       003609
C  [RES3K1(17,2)]*(U**2) + [RES3K2(17,2)]*(U) + [RES3K3(17,2)]          003610
C                                                                       003611
      V2(1)=VN1(1)+VN2(1)+VN3(1)+VN4(1)                                 003612
      V2(2)=VN1(2)+VN2(2)+VN3(2)+VN4(2)                                 003613
      V3(1)=VN5(1)+VN6(1)+VN7(1)+VN8(1)+VN9(1)                          003614
      V3(2)=VN5(2)+VN6(2)+VN7(2)+VN8(2)+VN9(2)                          003615
C                                                                       003616
      RES3K1(JRES3,1)=2.0*DXAP*(V2(1)+V3(1)+VN10(1))                    003617
      RES3K1(JRES3,2)=2.0*DXAP*(V2(2)+V3(2)+VN10(2))                    003618
C                                                                       003619
      V1(1)=-2.0*RES3POL(JRES3,1)                                       003620
      V1(2)=-2.0*RES3POL(JRES3,2)                                       003621
      CALL MULT(V1(1),V1(2),V2(1),V2(2),V4(1),V4(2))                    003622
      V1(1)=-RES3POL(JRES3,1)-DXA                                       003623
      V1(2)=-RES3POL(JRES3,2)                                           003624
      CALL MULT(V1(1),V1(2),V3(1),V3(2),V5(1),V5(2))                    003625
      RES3K2(JRES3,1)=2.0*DXAP*(V4(1)+V5(1)-2.0*DXA*VN10(1))            003626
      RES3K2(JRES3,2)=2.0*DXAP*(V4(2)+V5(2)-2.0*DXA*VN10(2))            003627
C                                                                       003628
      V1(1)=-RES3POL(JRES3,1)                                           003629
      V1(2)=-RES3POL(JRES3,2)                                           003630
```

```
      V4(1)=V1(1)                                                          003631
      V4(2)=V1(2)                                                          003632
      CALL MULT(V1(1),V1(2),V4(1),V4(2),V5(1),V5(2))                       003633
      CALL MULT(V5(1),V5(2),V2(1),V2(2),V1(1),V1(2))                       003634
      V4(1)=DXA*RES3POL(JRES3,1)                                           003635
      V4(2)=DXA*RES3POL(JRES3,2)                                           003636
      CALL MULT(V4(1),V4(2),V3(1),V3(2),V5(1),V5(2))                       003637
      RES3K3(JRES3,1)=2.0*DXAP*(V1(1)+V5(1)+DXA*DXA*UN10(1))               003638
      RES3K3(JRES3,2)=2.0*DXAP*(V1(2)+V5(2)+DXA*DXA*UN10(2))               003639
C                                                                          003640
      RETURN                                                               003641
      END                                                                  003642
C                                                                          003643
C                                                                          003644
C------------------------------------------------------------------------- 003645
CDECK RES1                                                                 003646
      OVERLAY(27,5)                                                        003647
      PROGRAM RES1                                                         003648
C------------------------------------------------------------------------- 003649
C                                                                          003650
C------------------------------------------------------------------------- 003651
C     SUBROUTINE RES1                                                      003652
C------------------------------------------------------------------------- 003653
C                                                                          003654
C                                                                          003655
C   THIS SUBROUTINE FORMS THE OVERALL LOW-RATE TRANSFER FUNCTION FROM      003656
C   THE RESIDUES FOR THE SIMPLE POLES BY ADDING THE INDIVIDUAL             003657
C   ZERO-ORDER  OVER  FIRST-ORDER RESIDUES AND STORING THE                 003658
C   RESULTS AS                                                           003659
C                                                                          003660
C         TF(Z) = Z * [RES1N(156) /RES1D(156)]                             003661
C         TF(W) = (W+1) * [RES1N(156)/RES1D(156)]                          003662
C         TF(W') = (W'+1) * [RES1N(156)/RES1D(156)]                        003663
C                                                                          003664
C   THE FOLLOWING MECHANIZATION SCHEME IS REPEATED JRES1                   003665
C   TIMES (NUMBER OF SIMPLE POLES).                                        003666
C     N/D + N1/D1 = (N*D1 + D*N1) / (D*D1)                                 003667
C                                                                          003668
C   THE INDIVIDUAL OPERATIONS USED TO EVALUATE THIS                        003669
C   ADDITION OF TRANSFER FUNCTIONS ARE:                                    003670
C         (1)   C3 = N*D1                                                  003671
C         (2)   N =  C3                                                    003672
C         (3)   C3 = D*N1                                                  003673
C         (4)   C6 = N + N1                                                003674
C         (5)   N = C6                                                     003675
C         (6)   C3 = D*D1                                                  003676
C         (7)   D = C3                                                     003677
C                                                                          003678
C   THE ARRAYS IN THE SUBROUTINE ASSOCIATED WITH THE                       003679
C   ABOVE OPERATIONS ARE:                                                  003680
C                                                                          003681
C         N = RES1N(156)                                                   003682
C         D = RES1D(156)                                                   003683
C        N1 = RES1NS(3)                                                    003684
C        D1 = RES1DS(6)                                                    003685
```

```
C         C3 = C3(318)                                                  003686
C         C6 = C6(318)                                                  003687
C                                                                       003688
C                                                                       003689
      DOUBLE RES1D(156),RES1N(156),RES1DS(6),RES1NS(3)                  003690
      DOUBLE A(106),B(106),C3(318),C6(318)                              003691
      DOUBLE RES1K(51,2),RES1POL(51,2)                                  003692
      DOUBLE XMAX,TINT,TEXT,XNT,DBCLK,DZERO,PWR                         003693
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,MM,NORDER                003694
      COMMON/ZEROS/KRES1Z,KRES2Z,KRES3Z                                 003695
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK         003696
      COMMON/TXCONV2/A,B,C3,C6                                          003697
      COMMON/TXCONV3/RES1K,RES1POL                                      003698
      COMMON/TXCONV6/RES1N,RES1D,KRES1N,KRES1D                          003699
C                                                                       003700
C  INITIALIZE ARRAYS TO ZERO                                            003701
      IORD=3*(NORDER+2)                                                 003702
      DO 100 I=1,IORD                                                   003703
      RES1N(I)=0.0                                                      003704
  100 RES1D(I)=0.0                                                      003705
C  SET INITIAL ELEMENT COUNT IN RES1N & RES1D                           003706
      KRES1N=3                                                          003707
      KRES1D=6                                                          003708
C  INITIALIZE OVERALL STORAGE ARRAYS WITH                               003709
C  FIRST RESIDUE TERM                                                   003710
      RES1D(1)=1.0                                                      003711
      RES1D(2)=0.0                                                      003712
      RES1D(3)=1.0                                                      003713
      RES1D(4)= -RES1POL(1,1)                                           003714
      RES1D(5)= -RES1POL(1,2)                                           003715
      RES1D(6)=0.0                                                      003716
      RES1N(1)=RES1K(1,1)                                               003717
      RES1N(2)=RES1K(1,2)                                               003718
      RES1N(3)=0.0                                                      003719
C                                                                       003720
      IF(JRES1 .EQ. 1) GO TO 250                                        003721
C                                                                       003722
C  LOOP TO ADD INDIVIDUAL RESIDUE TERMS                                 003723
C  JRES1 = NUMBER OF SIMPLE POLES                                       003724
      DO 200 K=2,JRES1                                                  003725
C  TEMPORARILY STORE NEXT RESIDUE TERM                                  003726
      RES1DS(1)=1.0                                                     003727
      RES1DS(2)=0.0                                                     003728
      RES1DS(3)=1.0                                                     003729
      RES1DS(4)= -RES1POL(K,1)                                          003730
      RES1DS(5)= -RES1POL(K,2)                                          003731
      RES1DS(6)=0.0                                                     003732
      RES1NS(1)=RES1K(K,1)                                              003733
      RES1NS(2)=RES1K(K,2)                                              003734
      RES1NS(3)=0.0                                                     003735
C  CALCULATE 1ST NUMERATOR TERM  C3=N*D1                                003736
C  RES1N = INPUT POLY                                                   003737
C  KRES1N = 2 * (RES1N ORDER + 1)                                       003738
C  RES1DS = 2ND INPUT POLY                                              003739
C  6 = 2 * (RES1DS ORDER + 1)                                           003740
```

```
C  C3 = OUTPUT POLY = RES1N*RES1DS                                    003741
C  NT3 = 2 * (C3 ORDER + 1)                                          003742
C  3 = FORMAT FOR INPUT POLYS : REAL,IMAG,Z-PWR                       003743
      CALL MULTIP(RES1N,KRES1N,RES1DS,6,C3,NT3,3)                     003744
C                                                                     003745
C  STORE 1ST NUMERATOR TERM :  N=C3                                   003746
      DO 300 I=1,NT3                                                  003747
  300 RES1N(I)=C3(I)                                                  003748
C  SET NEW ORDER OF FIRST NUMERATOR TERM                              003749
      KRES1N=NT3                                                      003750
C  CALCULATE 2ND NUMERTATOR TERM:  C3=D*N1                            003751
      CALL MULTIP(RES1D,KRES1D,RES1NS,3,C3,NT3,3)                     003752
C                                                                     003753
C  ADD 1ST & 2ND NUMERATOR TERMS                                      003754
C  RES1N = INPUT POLY                                                 003755
C  KRES1N = 2 * (RES1N ORDER + 1)                                     003756
C  C3 = INPUT POLY                                                    003757
C  NT3 = 2 * (C3 ORDER + 1)                                           003758
C  C6 = OUTPUT POLY = RES1N+C3                                        003759
C  NT6 = 2 * (C6 ORDER + 1)                                           003760
C  3 = FORMAT FOR INPUT POLYS: REAL,IMAG,Z-PWR                        003761
      CALL ADD(RES1N,KRES1N,C3,NT3,C6,NT6,3)                          003762
C                                                                     003763
C  STORE TOTAL NUMERATOR POLY:  N=C6                                  003764
      DO 400 I=1,NT6                                                  003765
  400 RES1N(I)=C6(I)                                                  003766
C  SET NEW ORDER OF TOTAL NUMERATOR POLY                              003767
      KRES1N=NT6                                                      003768
C  CALCULATE TOTAL DENOMINATOR POLY:  C3=D*D1                         003769
      CALL MULTIP(RES1D,KRES1D,RES1DS,6,C3,NT3,3)                     003770
C                                                                     003771
C  STORE TOTAL DENOMINATOR POLY:  D=C3                                003772
      DO 500 I=1,NT3                                                  003773
  500 RES1D(I)=C3(I)                                                  003774
C  SET NEW ORDER OF TOTAL DENOMINATOR POLY                            003775
      KRES1D=NT3                                                      003776
  200 CONTINUE                                                        003777
C                                                                     003778
  250 CONTINUE                                                        003779
C                                                                     003780
C  ORDER NUMERATOR POLY IN DESCENDING PWRS OF Z, W, OR W'             003781
      CALL ORDER3(RES1N,KRES1N,3)                                     003782
C                                                                     003783
C     FIND LARGEST NUMERATOR COEFFICIENT                              003784
      IF(KRES1N .GT. 0) GO TO 700                                     003785
      XMAX=0.0                                                        003786
      DO 600 I=1,KRES1N,3                                             003787
  600 IF(DABS(RES1N(I)) .GT. XMAX) XMAX = DABS(RES1N(I))              003788
C                                                                     003789
C  ELIMINATE EXTREMELY SMALL HIGHEST ORDER NUM COEFF TERM             003790
      IF((DABS(RES1N(1))/XMAX) .LT. 1.D-12) GO TO 710                 003791
      GO TO 700                                                       003792
  710 RES1N(1)=0.0                                                    003793
      RES1N(2)=0.0                                                    003794
      RES1N(3)=0.0                                                    003795
```

```
C  CALL SIMPLE ROUTINE TO PUSH ZEROED                                    003796
C  OUT TERM TO BOTTOM OF ARRAY KRESIN                                    003797
      CALL SIMPLE(RES1N,KRES1N,3)                                        003798
  700 CONTINUE                                                           003799
C                                                                        003800
C  ***************  ADD ZERO OR CANCEL POLE  ***********************    003801
C  **********  AT  Z=0.0,  W=-1.0,  OR  W'=-2/TEXT  ************        003802
C                                                                        003803
C  THIS SECTION HUNTS AND CANCELS DENOMINATOR POLE AT Z=0.0              003804
C  W=-1.0, OR W'=-2/TEXT INSTEAD OF ADDING LIKE ZERO TO THE              003805
C  NUMERATOR.                                                            003806
C                                                                        003807
      KSKIP=0                                                            003808
      ZERO=0.0                                                           003809
      DZERO=0.0                                                          003810
      IF(TXFORM .EQ. 2HZT) GO TO 715                                     003811
      ZERO=-1.0                                                          003812
      DZERO=-1.0                                                         003813
      IF(TXFORM .EQ. 2HUP) ZERO=-2.0/TEXT                                003814
      IF(TXFORM .EQ. 2HUP) DZERO=-2.0/TEXT                               003815
  715 CONTINUE                                                           003816
      DO 720 I=1,JRES1                                                   003817
      ISH=I                                                              003818
      P1=RES1POL(I,1)                                                    003819
      P2=RES1POL(I,2)                                                    003820
      IF(ABS(P1-ZERO).LE.1.E-11.AND.ABS(P2).LE.1.E-11) GO TO 725         003821
  720 CONTINUE                                                           003822
      GO TO 730                                                          003823
C  SET FLAG TO INDICATE THAT POLE HAS BEEN CANCELLED                     003824
  725 KSKIP=10                                                           003825
C  CANCEL POLE BY OVER WRITTING IT                                       003826
      DO 735 J=ISH,JRES1                                                 003827
      RES1POL(J,1)=RES1POL(J+1,1)                                        003828
  735 RES1POL(J,2)=RES1POL(J+1,2)                                        003829
      JRES1=JRES1-1                                                      003830
  730 CONTINUE                                                           003831
C  MULTIPLY NUMERATOR POLY IN RES1N(156) BY  (Z) OR (W+A)                003832
C  IF NO LIKE POLE IS FOUND                                              003833
      IF(KSKIP .GT. 5) GO TO 740                                         003834
      A(1)=1.0                                                           003835
      A(2)=0.0                                                           003836
      A(3)=1.0                                                           003837
      A(4)=-DZERO                                                        003838
      A(5)=0.0                                                           003839
      A(6)=0.0                                                           003840
      CALL MULTIP(A,6,RES1N,KRES1N,C3,NT3,3)                             003841
C  TRANSFER NEW NUMERATOR POLY INTO RES1N(156)                           003842
      DO 745 I=1,NT3                                                     003843
  745 RES1N(I)=C3(I)                                                     003844
      KRES1N=NT3                                                         003845
  740 CONTINUE                                                           003846
C  ORDER NEW NUMERATOR POLY IN DESCENDING PWRS OF Z, W, OR W'            003847
      CALL ORDER3(RES1N,KRES1N,3)                                        003848
C  TRANSFER REAL AND IMAGINARY PART OF NUMERATOR COEFFICIENTS            003849
C  IN ARRAY RES1N INTO TWO SEPARATE ARRAYS 'A' AND 'B'                   003850
```

```
      K=1                                                               003851
      DO 755 I=1,KRES1N,3                                               003852
      A(K)=RES1N(I)                                                     003853
      B(K)=RES1N(I+1)                                                   003854
  755 K=K+1                                                             003855
C  SET DEGREE OF NUMERATOR POLY AND NUMBER OF ZEROS IN KRES1Z          003856
      KRES1Z=RES1N(3)                                                   003857
C  OBTAIN NUMERATOR ROOTS                                               003858
      IF(KRES1Z .EQ. 0) GO TO 762                                       003859
      CALL ROOTS(A,B,KRES1Z,C3,C6)                                      003860
C  TRANSFER NUMERATOR ROOTS INTO ARRAY RES1K(51,2)                      003861
      DO 760 I=1,KRES1Z                                                 003862
      RES1K(I,1)=C3(I)                                                  003863
  760 RES1K(I,2)=C6(I)                                                  003864
  762 CONTINUE                                                          003865
C  FORM NEW DENOMINATOR POLYNOMIAL IF POLE CANCELLED                    003866
      IF(KSKIP .LT. 5) GO TO 765                                        003867
C  TRANSFER REAL AND IMAGINARY PART OF POLES INTO                       003868
C  ARRAY 'A' IN SEQUENTIAL BACK-TO-BACK FORMAT                          003869
      RES1D(1)=1.0                                                      003870
      RES1D(2)=0.0                                                      003871
      RES1D(3)=0.0                                                      003872
      KRES1D=3                                                          003873
      IF(JRES1 .EQ. 0) GO TO 765                                        003874
      DO 770 I=1,JRES1                                                  003875
      A(I)=RES1POL(I,1)                                                 003876
  770 B(I)=RES1POL(I,2)                                                 003877
C  CALL SUBROUTINE 'COMPOLY' TO OBTAIN NEW DENOMINATOR POLY             003878
      CALL COMPOLY(C3,C6,A,B,JRES1)                                     003879
C  TRANSFER DENOMINATOR POLY COEFFS INTO ARRAY RES1D(156)               003880
C  AND ADD POLY POWERS.                                                 003881
      K=1                                                               003882
      N=JRES1+1                                                         003883
      DO 771 I=1,N                                                      003884
      PWR=FLOAT(JRES1-I+1)                                              003885
      RES1D(K)=C3(I)                                                    003886
      RES1D(K+1)=C6(I)                                                  003887
      RES1D(K+2)=PWR                                                    003888
  771 K=K+3                                                             003889
      KRES1D=K-1                                                        003890
  765 CONTINUE                                                          003891
      RETURN                                                            003892
      END                                                               003893
C                                                                       003894
C-----------------------------------------------------------------------003895
C                                                                       003896
C-----------------------------------------------------------------------003897
CDECK RES2                                                              003898
      OVERLAY(27,6)                                                     003899
      PROGRAM RES2                                                      003900
C-----------------------------------------------------------------------003901
C                                                                       003902
C-----------------------------------------------------------------------003903
C     SUBROUTINE RES2                                                   003904
C-----------------------------------------------------------------------003905
C
```

```
C                                                                       003906
C     THIS SUBROUTINE FORMS THE OVERALL U OR U' TRANSFER FUNCTION       003907
C     FROM THE RESIDUES FOR POLES WITH MULTIPLICITY EQUAL TO TWO.       003908
C     THE INDIVIDUAL FIRST-ORDER OVER SECOND-ORDER RESIDUES ARE         003909
C     ADDED BY FORMING THE NUMERATOR-DENOMINATOR CROSS-PRODUCTS         003910
C     TO OBTAIN THE COMBINED THE NUMERATOR AND THEN MULTIPLYING         003911
C     THE DENOMINATOR TERMS TO FORM THE COMBINED DENOMINATOR.           003912
C                                                                       003913
C            N/D + N1/D1 = (N*D1 + D*N1)/(D*D1)                         003914
C                                                                       003915
C     THE MECHANIZATION SCHEME USED IS AS FOLLOWS:                      003916
C                                                                       003917
C     [A+U] * [(RES2K1)*U + (RES2K2)]   =  NUMERATOR                    003918
C     [(U**2) + (2*RES2POL)*U + (RES2POL**2)]  =  DENOMINATOR           003919
C     NOTE:  RES2POL = -RES2POL                                         003920
C                                                                       003921
C                                                                       003922
      DOUBLE RES2D(156),RES2N(156),RES2DS(9),RES2NS(6)                  003923
      DOUBLE A(106),B(106),C3(318),C6(318)                              003924
      DOUBLE RES2K1(25,2),RES2K2(25,2),RES2POL(25,2)                    003925
      DOUBLE XMAX,TINT,TEXT,XNT,DZERO,R(2),X,Y,DBCLK                    003926
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER                003927
      COMMON/ZEROS/KRES1Z,KRES2Z,KRES3Z                                 003928
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK         003929
      COMMON/TXCONV2/A,B,C3,C6                                          003930
      COMMON/TXCONV4/RES2K1,RES2K2,RES2POL                              003931
      COMMON/TXCONV7/RES2N,RES2D,KRES2N,KRES2D                          003932
C                                                                       003933
C                                                                       003934
C  INITIALIZE ARRAYS TO ZERO                                            003935
      IORD=3*(NORDER+2)                                                 003936
      DO 100 I=1,IORD                                                   003937
      RES2N(I)=0.0                                                      003938
  100 RES2D(I)=0.0                                                      003939
C                                                                       003940
C  SET INITIAL ELEMENT COUNT IN RES2N AND RES2D                         003941
      KRES2N=6                                                          003942
      KRES2D=9                                                          003943
C                                                                       003944
C  INITIALIZE OVERALL STORAGE ARRAYS WITH FIRST RESIDUE TERM            003945
      RES2N(1)=RES2K1(1,1)                                              003946
      RES2N(2)=RES2K1(1,2)                                              003947
      RES2N(3)=1.0                                                      003948
      RES2N(4)=RES2K2(1,1)                                              003949
      RES2N(5)=RES2K2(1,2)                                              003950
      RES2N(6)=0.0                                                      003951
C                                                                       003952
      R(1)=RES2POL(1,1)                                                 003953
      R(2)=RES2POL(1,2)                                                 003954
      CALL MULT(R(1),R(2),R(1),R(2),X,Y)                                003955
C                                                                       003956
      RES2D(1)=1.0                                                      003957
      RES2D(2)=0.0                                                      003958
      RES2D(3)=2.0                                                      003959
      RES2D(4)= -2.0*RES2POL(1,1)                                       003960
```

```
      RES2D(5)= -2.0*RES2POL(1,2)                                       003961
      RES2D(6)=1.0                                                      003962
      RES2D(7)=X                                                        003963
      RES2D(8)=Y                                                        003964
      RES2D(9)=0.0                                                      003965
C                                                                       003966
      IF(JRES2 .EQ. 1) GO TO 250                                        003967
C                                                                       003968
C  LOOP TO ADD INDIVDUAL RESIDUE TERMS                                  003969
C  JRES2 = NUMBER OF SECOND ORDER POLES                                 003970
      DO 200 K=2,JRES2                                                  003971
C  TEMPORARILY STORE NEXT RESIDUE TERM IN RES2NS AND RES2DS             003972
      RES2NS(1)=RES2K1(K,1)                                             003973
      RES2NS(2)=RES2K1(K,2)                                             003974
      RES2NS(3)=1.0                                                     003975
      RES2NS(4)=RES2K2(K,1)                                             003976
      RES2NS(5)=RES2K2(K,2)                                             003977
      RES2NS(6)=0.0                                                     003978
C                                                                       003979
      R(1)=RES2POL(K,1)                                                 003980
      R(2)=RES2POL(K,2)                                                 003981
      CALL MULT(R(1),R(2),R(1),R(2),X,Y)                                003982
C                                                                       003983
      RES2DS(1)=1.0                                                     003984
      RES2DS(2)=0.0                                                     003985
      RES2DS(3)=2.0                                                     003986
      RES2DS(4)= -2.0*RES2POL(K,1)                                      003987
      RES2DS(5)= -2.0*RES2POL(K,2)                                      003988
      RES2DS(6)=1.0                                                     003989
      RES2DS(7)=X                                                       003990
      RES2DS(8)=Y                                                       003991
      RES2DS(9)=0.0                                                     003992
C                                                                       003993
C  CALCULTE:   C3=N*D1                                                  003994
      CALL MULTIP(RES2N,KRES2N,RES2DS,9,C3,NT3,3)                       003995
      DO 300 I=1,NT3                                                    003996
  300 RES2N(I)=C3(I)                                                    003997
      KRES2N=NT3                                                        003998
C                                                                       003999
C  CALCULTE:   C3=D*N1                                                  004000
      CALL MULTIP(RES2D,KRES2D,RES2NS,6,C3,NT3,3)                       004001
C  ADD (N*D1) + (D*N1)                                                  004002
      CALL ADD(RES2N,KRES2N,C3,NT3,C6,NT6,3)                            004003
      DO 400 I=1,NT6                                                    004004
  400 RES2N(I)=C6(I)                                                    004005
      KRES2N=NT6                                                        004006
C                                                                       004007
C  CALCULATE DENOMINATOR POLY:   C3=D*D1                                004008
      CALL MULTIP(RES2D,KRES2D,RES2DS,9,C3,NT3,3)                       004009
      DO 500 I=1,NT3                                                    004010
  500 RES2D(I)=C3(I)                                                    004011
      KRES2D=NT3                                                        004012
C                                                                       004013
  200 CONTINUE                                                          004014
  250 CONTINUE                                                          004015
```

```
C                                                                       004016
      CALL ORDER3(RES2N,KRES2N,3)                                       004017
C                                                                       004018
C  ELIMINATE EXTREMELY SMALL HIGHEST-ORDER NUM COEFF TERM               004019
      IF(KRES2N .GT. 0) GO TO 700                                       004020
      XMAX=0.0                                                          004021
      DO 600 I=1,KRES2N,3                                               004022
  600 IF(DABS(RES2N(I)) .GT. XMAX) XMAX=DABS(RES2N(I))                  004023
      IF((DABS(RES2N(1))/XMAX) .LT. 1.D-12) GO TO 710                   004024
      GO TO 700                                                         004025
  710 RES2N(1)=0.0                                                      004026
      RES2N(2)=0.0                                                      004027
      RES2N(3)=0.0                                                      004028
      CALL SIMPLE(RES2N,KRES2N,3)                                       004029
  700 CONTINUE                                                          004030
C                                                                       004031
C  MULTIPLE NUMERATOR POLY IN RES2N(156) BY  (A+U)                      004032
      DZERO=1.0                                                         004033
      IF(TXFORM .EQ. 2HUP) DZERO=2.0/TEXT                               004034
      A(1)=1.0                                                          004035
      A(2)=0.0                                                          004036
      A(3)=1.0                                                          004037
      A(4)=DZERO                                                        004038
      A(5)=0.0                                                          004039
      A(6)=0.0                                                          004040
      CALL MULTIP(A,6,RES2N,KRES2N,C3,NT3,3)                            004041
      DO 745 I=1,NT3                                                    004042
  745 RES2N(I)=C3(I)                                                    004043
      KRES2N=NT3                                                        004044
C                                                                       004045
C  OBTAIN NUMERATOR ROOTS                                               004046
      CALL ORDER3(RES2N,KRES2N,3)                                       004047
      K=1                                                               004048
      DO 755 I=1,KRES2N,3                                               004049
      A(K)=RES2N(I)                                                     004050
      B(K)=RES2N(I+1)                                                   004051
  755 K=K+1                                                             004052
      KRES2Z=RES2N(3)                                                   004053
      IF(KRES2Z .EQ. 0) GO TO 770                                       004054
      CALL ROOTS(A,B,KRES2Z,C3,C6)                                      004055
C                                                                       004056
C  TRANSFER NUMERATOR ROOTS INTO ARRAYS  RES2K1(25,2)                   004057
C  AND RES2K2(25,2)                                                     004058
      KZ=KRES2Z                                                         004059
      IF(KRES2Z .GT. 25) KZ=25                                          004060
      DO 760 I=1,KZ                                                     004061
      RES2K1(I,1)=C3(I)                                                 004062
  760 RES2K1(I,2)=C6(I)                                                 004063
      IF(KRES2Z .GT. 25) GO TO 765                                      004064
      GO TO 770                                                         004065
  765 KZ=KRES2Z-25                                                      004066
      DO 775 I=1,KZ                                                     004067
      RES2K2(I,1)=C3(I+26)                                              004068
  775 RES2K2(I,2)=C6(I+26)                                              004069
  770 CONTINUE                                                          004070
```

```
C                                                                       004071
      RETURN                                                            004072
      END                                                               004073
C                                                                       004074
C                                                                       004075
C-----------------------------------------------------------------------004076
CDECK RES3                                                              004077
      OVERLAY(27,7)                                                     004078
      PROGRAM RES3                                                      004079
C-----------------------------------------------------------------------004080
C                                                                       004081
C-----------------------------------------------------------------------004082
C     SUBROUTINE RES3                                                   004083
C-----------------------------------------------------------------------004084
C                                                                       004085
C                                                                       004086
C     THIS SUBROUTINE FORMS THE OVERALL U OR U' TRANSFER FUNCTION       004087
C     FROM THE RESIDUES FOR POLES WITH MULTIPLICITY EQUAL TO THREE.     004088
C     THE INDIVIDUAL SECOND-ORDER OVER THIRD-ORDER RESIDUES ARE         004089
C     ADDED BY FORMING THE NUMERATOR-DENOMINATOR CROSS-PRODUCTS         004090
C     TO OBTAIN THE COMBINED THE NUMERATOR AND THEN MULTIPLYING         004091
C     THE DENOMINATOR TERMS TO FORM THE COMBINED DENOMINATOR.           004092
C                                                                       004093
C           N/D + N1/D1 = (N*D1 + D*N1)/(D*D1)                          004094
C                                                                       004095
C     THE MECHANIZATION SCHEME USED IS AS FOLLOWS:                      004096
C                                                                       004097
C     [A+U] * [(RES3K1)*(U**2) + (RES3K2)*U + (RES3K3)] /               004098
C     [(U**3) + (3*P)*(U**2) + (3*P*P)*(U) + (P**3)]                    004099
C          P = -RES3POL                                                 004100
C                                                                       004101
C                                                                       004102
      DOUBLE RES3D(156),RES3N(156),RES3DS(12),RES3NS(9)                 004103
      DOUBLE A(106),B(106),C3(318),C6(318)                              004104
      DOUBLE RES3K1(17,2),RES3K2(17,2),RES3K3(17,2),RES3POL(17,2)       004105
      DOUBLE XMAX,DZERO,R1(2),R2(2),R3(2),TINT,TEXT,XNT,DBCLK           004106
      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,MM,NORDER                004107
      COMMON/ZEROS/KRES1Z,KRES2Z,KRES3Z                                 004108
      COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK         004109
      COMMON/TXCONU2/A,B,C3,C6                                          004110
      COMMON/TXCONU5/RES3K1,RES3K2,RES3K3,RES3POL                       004111
      COMMON/TXCONU8/RES3N,RES3D,KRES3N,KRES3D                          004112
C                                                                       004113
C                                                                       004114
C  INITIALIZE ARRAYS TO ZERO                                            004115
      IORD=3*(NORDER+2)                                                 004116
      DO 100 I=1,IORD                                                   004117
      RES3N(I)=0.0                                                      004118
  100 RES3D(I)=0.0                                                      004119
C                                                                       004120
C  SET INITIAL ELEMENT COUNT IN RES3N AND RES3D                         004121
      KRES3N=9                                                          004122
      KRES3D=12                                                         004123
C                                                                       004124
C  INITIALIZE OVERALL STORAGE ARRAYS WITH FIRST RESIDUE TERM            004125
```

```
      RES3N(1)=RES3K1(1,1)                                              004126
      RES3N(2)=RES3K1(1,2)                                              004127
      RES3N(3)=2.0                                                      004128
      RES3N(4)=RES3K2(1,1)                                              004129
      RES3N(5)=RES3K2(1,2)                                              004130
      RES3N(6)=1.0                                                      004131
      RES3N(7)=RES3K3(1,1)                                              004132
      RES3N(8)=RES3K3(1,2)                                              004133
      RES3N(9)=0.0                                                      004134
C                                                                       004135
      R1(1)=-RES3POL(1,1)                                               004136
      R1(2)=-RES3POL(1,2)                                               004137
      CALL MULT(R1(1),R1(2),R1(1),R1(2),R2(1),R2(2))                    004138
      CALL MULT(R2(1),R2(2),R1(1),R1(2),R3(1),R3(2))                    004139
C                                                                       004140
      RES3D(1)=1.0                                                      004141
      RES3D(2)=0.0                                                      004142
      RES3D(3)=3.0                                                      004143
      RES3D(4)= 3.0*R1(1)                                               004144
      RES3D(5)= 3.0*R1(2)                                               004145
      RES3D(6)=2.0                                                      004146
      RES3D(7)=3.0*R2(1)                                                004147
      RES3D(8)=3.0*R2(2)                                                004148
      RES3D(9)=1.0                                                      004149
      RES3D(10)=R3(1)                                                   004150
      RES3D(11)=R3(2)                                                   004151
      RES3D(12)=0.0                                                     004152
C                                                                       004153
      IF(JRES3 .EQ. 1) GO TO 250                                        004154
C                                                                       004155
C  LOOP TO ADD INDIVIDUAL RESIDUE TERMS                                 004156
C  JRES3 = NUMBER OF SECOND ORDER POLES                                 004157
      DO 200 K=2,JRES3                                                  004158
C  TEMPORARILY STORE NEXT RESIDUE TERM IN RES3NS AND RES3DS             004159
      RES3NS(1)=RES3K1(K,1)                                             004160
      RES3NS(2)=RES3K1(K,2)                                             004161
      RES3NS(3)=2.0                                                     004162
      RES3NS(4)=RES3K2(K,1)                                             004163
      RES3NS(5)=RES3K2(K,2)                                             004164
      RES3NS(6)=1.0                                                     004165
      RES3NS(7)=RES3K3(K,1)                                             004166
      RES3NS(8)=RES3K3(K,2)                                             004167
      RES3NS(9)=0.0                                                     004168
C                                                                       004169
      R1(1)= -RES3POL(K,1)                                              004170
      R1(2)= -RES3POL(K,2)                                              004171
      CALL MULT(R1(1),R1(2),R1(1),R1(2),R2(1),R2(2))                    004172
      CALL MULT(R2(1),R2(2),R1(1),R1(2),R3(1),R3(2))                    004173
C                                                                       004174
      RES3DS(1)=1.0                                                     004175
      RES3DS(2)=0.0                                                     004176
      RES3DS(3)=3.0                                                     004177
      RES3DS(4)= 3.0*R1(1)                                              004178
      RES3DS(5)= 3.0*R1(2)                                              004179
      RES3DS(6)=2.0                                                     004180
```

```
      RES3DS(7)=3.0*R2(1)                                               004181
      RES3DS(8)=3.0*R2(2)                                               004182
      RES3DS(9)=1.0                                                     004183
      RES3DS(10)=R3(1)                                                  004184
      RES3DS(11)=R3(2)                                                  004185
      RES3DS(12)=0.0                                                    004186
C                                                                       004187
C  CALCULTE:   C3=N*D1                                                  004188
      CALL MULTIP(RES3N,KRES3N,RES3DS,12,C3,NT3,3)                      004189
      DO 300 I=1,NT3                                                    004190
  300 RES3N(I)=C3(I)                                                    004191
      KRES3N=NT3                                                        004192
C                                                                       004193
C  CALCULTE:   C3=D*N1                                                  004194
      CALL MULTIP(RES3D,KRES3D,RES3NS,9,C3,NT3,3)                       004195
C  ADD (N*D1) + (D*N1)                                                  004196
      CALL ADD(RES3N,KRES3N,C3,NT3,C6,NT6,3)                            004197
      DO 400 I=1,NT6                                                    004198
  400 RES3N(I)=C6(I)                                                    004199
      KRES3N=NT6                                                        004200
C                                                                       004201
C  CALCULATE DENOMINATOR POLY:   C3=D*D1                                004202
      CALL MULTIP(RES3D,KRES3D,RES3DS,12,C3,NT3,3)                      004203
      DO 500 I=1,NT3                                                    004204
  500 RES3D(I)=C3(I)                                                    004205
      KRES3D=NT3                                                        004206
C                                                                       004207
  200 CONTINUE                                                          004208
  250 CONTINUE                                                          004209
C                                                                       004210
      CALL ORDER3(RES3N,KRES3N,3)                                       004211
C                                                                       004212
C  ELIMINATE EXTREMELY SMALL HIGHEST-ORDER NUM COEFF TERM               004213
      IF(KRES3N .GT. 0) GO TO 700                                       004214
      XMAX=0.0                                                          004215
      DO 600 I=1,KRES3N,3                                               004216
  600 IF(DABS(RES3N(I)) .GT. XMAX) XMAX=DABS(RES3N(I))                  004217
      IF((DABS(RES3N(1))/XMAX) .LT. 1.D-12) GO TO 710                   004218
      GO TO 700                                                         004219
  710 RES3N(1)=0.0                                                      004220
      RES3N(2)=0.0                                                      004221
      RES3N(3)=0.0                                                      004222
      CALL SIMPLE(RES3N,KRES3N,3)                                       004223
  700 CONTINUE                                                          004224
C                                                                       004225
C  MULTIPLE NUMERATOR POLY IN RES3N(156) BY  (A+U)                      004226
      DZERO=1.0                                                         004227
      IF(TXFORM .EQ. 2HUP) DZERO=2.0/TEXT                               004228
      A(1)=1.0                                                          004229
      A(2)=0.0                                                          004230
      A(3)=1.0                                                          004231
      A(4)=DZERO                                                        004232
      A(5)=0.0                                                          004233
      A(6)=0.0                                                          004234
      CALL MULTIP(A,6,RES3N,KRES3N,C3,NT3,3)                            004235
```

```
      DO 745 I=1,NT3                                                    004236
  745 RES3N(I)=C3(I)                                                    004237
      KRES3N=NT3                                                        004238
C                                                                       004239
C  OBTAIN NUMERATOR ROOTS                                               004240
      CALL ORDER3(RES3N,KRES3N,3)                                       004241
      K=1                                                               004242
      DO 755 I=1,KRES3N,3                                               004243
      A(K)=RES3N(I)                                                     004244
      B(K)=RES3N(I+1)                                                   004245
  755 K=K+1                                                             004246
      KRES3Z=RES3N(3)                                                   004247
      IF(KRES3Z .EQ. 0) GO TO 770                                       004248
      CALL ROOTS(A,B,KRES3Z,C3,C6)                                      004249
C                                                                       004250
C  TRANSFER NUMERATOR ROOTS INTO ARRAYS  RES3K1(17,2)                   004251
C  RES3K2(17,2), AND RES3K3(17,2)                                       004252
      KZ=KRES3Z                                                         004253
      IF(KRES3Z .GT. 17) KZ=17                                          004254
      DO 760 I=1,KZ                                                     004255
      RES3K1(I,1)=C3(I)                                                 004256
  760 RES3K1(I,2)=C6(I)                                                 004257
      IF(KRES3Z .GT. 17) GO TO 765                                      004258
      GO TO 770                                                         004259
  765 KZ=KRES3Z-17                                                      004260
      IF(KZ .GT. 17) KZ=17                                              004261
      DO 775 I=1,KZ                                                     004262
      RES3K2(I,1)=C3(I+17)                                              004263
  775 RES3K2(I,2)=C6(I+17)                                              004264
      IF(KRES3Z .GT. 34) GO TO 766                                      004265
      GO TO 770                                                         004266
  766 KZ=KRES3Z-34                                                      004267
      DO 777 I=1,KZ                                                     004268
      RES3K3(I,1)=C3(I+34)                                              004269
  777 RES3K3(I,2)=C6(I+34)                                              004270
  770 CONTINUE                                                          004271
C                                                                       004272
      RETURN                                                            004273
      END                                                               004274
```


U.S.Government Printing Office: 1981 — 757-002/437


DATE
FILMED
8